FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

# Extracting medical information from personal child health records

**Fábio André da Silva Amarante**


Mestrado Integrado em Engenharia Informática e Computação

Supervisor: Rui Camacho

Second Supervisor: Luís Filipe Teixeira

February 19, 2018

# Extracting medical information from personal child health records

**Fábio André da Silva Amarante**

Mestrado Integrado em Engenharia Informática e Computação

Approved in oral examination by the committee:

Chair: Doctor Rosaldo Rossetti
External Examiner: Doctor Brígida Faria
Supervisor: Doctor Rui Camacho
February 19, 2018

# Abstract

Recent evolution on technology is providing more tools to different areas, such as Medicine. Handwritten medical records are a quite valuable asset of information. Extracting medical information from those records will help the development/improvement of knowledge-based systems, assisting medical doctors to make better decisions.

However, the large number of such records will require a considerable amount of working hours to extract manually and would be pruned to transcription errors. In Portugal, for example, the period from birth birth to age 18 is monitored and registered in individual health booklets. Information registered in those books include free handwritten text, numbers written in table's cells and charts. The information available in these records is valuable for several areas of study: pediatrics, nutrition, population studies, etc. Research in these areas require large amounts of data to have statistical support in order to suggest medical procedures acceptable to the community.

The main problem associated with this resides in the task of manually convert the health records into digital health records, which is a waste of time and resources. Thereby, applying Image Processing techniques, along with Optical Character Recognition (OCR), Machine Learning Classification and Text Mining it's possible to automate the extraction and conversion of traditional health records into digital format.

Providing automation of the entire process of extracting information contained in child and youth health books facilitate and improve the quality of research work in Pediatrics. The data will be provided in a platform that allows a quick and accurate analysis of information concerning one or more children.

The outcomes obtained allowed to conclude that although it is possible to idealize a solution that automates the entire process of extracting and converting information, it is necessary to improve and complete some of the work done in this dissertation.

# Resumo

A recente evolução da tecnologia está a fornecer mais ferramentas para diferentes áreas, como a Medicina. Os registos médicos manuscritos são um bem valioso de informação. A extração de informações médicas desses registos ajudará o desenvolvimento/aperfeiçoamento de sistemas baseados no conhecimento, auxiliando os médicos a tomar melhores decisões.

No entanto, um grande número desses registos exigirá uma quantidade considerável de horas de trabalho para extrair manualmente e estará sujeita a erros de transcrição. Em Portugal, por exemplo, o período desde o nascimento até aos 18 anos é monitorizado e registado em boletins de saúde individuais. As informações registadas nesses livros incluem texto manuscrito livre, números escritos em células de tabelas e gráficos. A informação disponível nestes registos é valiosa para várias áreas de estudo como: pediatria, nutrição, estudos populacionais, etc. A pesquisa nestas áreas requerem grandes quantidades de dados para obter suporte estatístico para sugerir procedimentos médicos aceitáveis para a comunidade.

O principal problema associado a isso reside na tarefa de converter manualmente os registos de saúde em registos de saúde digitais, que é um desperdício de tempo e recursos. Desta forma, aplicando técnicas de processamento de imagem, juntamente com Reconhecimento Ótico de Caracteres (OCR), classificação de "Machine Learning" e Mineração de Texto é possível automatizar a extração e conversão de registos de saúde tradicionais em formato digital.

Fornecer a automação de todo o processo de extração de informações contidas nos boletins de saúde infantil e juvenil facilita e melhora a qualidade do trabalho de pesquisa em Pediatria. Os dados serão fornecidos em uma plataforma que permite uma análise rápida e precisa das informações relativas a uma ou mais crianças.

Os resultados obtidos permitem concluir que embora seja possível idealizar uma solução que automatize todo o processo de extração e conversão de informação, é necessário melhorar e concluir algum do trabalho feito nesta dissertação.

# Acknowledgements

I would like to begin by thanking the entire family that provided me and supported me throughout the academic journey, either in economic conditions as in emotional situations.

I would like to thank my supervisor, professor Rui Camacho, and my co-supervisor, professor Luís Teixeira, for all the help, availability, sympathy, guidance and encouragement that I received during the development of the Dissertation. I would also like to thank INESC-TEC for its support throughout the project.

To my friends, and especially my girlfriend, a big thank you for all the support and time lost in motivating me and making me believe that it was possible. Indeed it was!

Finally, I would like to thank the project **"NanoSTIMA: Macro-to-Nano Human Sensing: Towards Integrated Multimodal Health Monitoring and Analytics/NORTE-01-0145-FEDER-000016"** funded by Northern Regional Operational Program (NORTE 2020), under the PORTUGAL 2020 Partnership Agreement, and through the European Regional Development Fund - ERDF for the provision of data used to carry out this project.

Fábio Amarante

# Contents


**1 Introduction** **1**
- 1.1 Context . . . 1
- 1.2 Problem description . . . 2
- 1.3 Motivation . . . 2
- 1.4 Goals . . . 2
- 1.5 Structure of the document . . . 2

**2 Image Processing and Data Mining** **5**
- 2.1 Digital Image Processing . . . 5
  - 2.1.1 Low-level categories . . . 6
  - 2.1.2 Mid-level categories . . . 7
  - 2.1.3 High-level categories . . . 12
  - 2.1.4 Available Tools . . . 13
- 2.2 Optical Character Recognition . . . 14
- 2.3 Data Mining . . . 15
  - 2.3.1 CRISP-DM . . . 16
  - 2.3.2 Tasks . . . 17
- 2.4 Text Mining . . . 19
  - 2.4.1 Text Mining Techniques . . . 20
- 2.5 Related Work . . . 22
- 2.6 Chapter Conclusions . . . 23

**3 Methodology and Implementation** **25**
- 3.1 Methodology . . . 25
  - 3.1.1 Application Flow . . . 25
- 3.2 Implementation . . . 28
  - 3.2.1 Page splitting . . . 28
  - 3.2.2 Table page . . . 29
  - 3.2.3 Chart page . . . 35
  - 3.2.4 Datasets . . . 40
  - 3.2.5 Tools and Libraries . . . 41

**4 Case study** **43**
- 4.1 Setup . . . 43
- 4.2 Results . . . 45
  - 4.2.1 Image processing . . . 45
  - 4.2.2 Classifier . . . 46

**5 Conclusions and Future Work** **49**
5.1 Objectives Fulfillment . . . . . 49
5.2 Future Work . . . . . 50

**References** **51**

# List of Figures


2.1 Image Processing Levels . . . . . 6
2.2 Image segmentation methods. [ZA15] . . . . . 8
2.3 Graph with weight and age of the child. . . . . 9
2.4 (a) Offline character recognition, (b) Online character recognition. . . . . 14
2.5 An example of a scanned page of an individual health booklet. (a) Printed text, (b) Free handwriting text, (c) Digits in a table. . . . . 15
2.6 Phases of the CRISP-DM model . . . . . 17
2.7 Steps on a text mining system. [FWRZ06] . . . . . 19
2.8 Usage of Text Mining tools. [KC16] . . . . . 20
2.9 Word spotting approaches on the last decade. [GSGN17] . . . . . 22

3.1 Application flowchart . . . . . 25
3.2 Preprocessing flowchart for each page . . . . . 26
3.3 Processing flowchart for table . . . . . 27
3.4 Processing flowchart for charts . . . . . 27
3.5 Consistency of information diagram . . . . . 28
3.6 Table image after removing white color . . . . . 30
3.7 Steps along the isolation of the characters on the table . . . . . 31
3.8 Steps along the detection of lines and columns of the table . . . . . 32
3.9 Steps along the segmentation of characters in a cell . . . . . 33
3.10 Characters after scaling and padding . . . . . 33
3.11 Chart boundaries detection . . . . . 36
3.12 Chart masks . . . . . 37
3.13 Identified chart axes . . . . . 37
3.14 Chart image without white . . . . . 38
3.15 Chart points detection steps . . . . . 39

4.1 Character grid feature . . . . . 44
4.2 Characters segmented from the table image . . . . . 45
4.3 Points identified on the chart image . . . . . 46

LIST OF FIGURES

# List of Tables


2.1 Comparison between image processing tools . . . . . 14
2.2 Recent studies . . . . . 23

4.1 Features and input size for classifiers . . . . . 44
4.2 Datasets dimensions for classifiers . . . . . 47
4.3 Training arguments size for NN . . . . . 47
4.4 Training arguments size for CNN . . . . . 47
4.5 Results for NN . . . . . 48
4.6 Results for CNN . . . . . 48

LIST OF TABLES

# Abbreviations

| | |
|---|---|
| ANN | Artifical Neural Network |
| API | Application Protocol Interface |
| BGR | Blue Green Red |
| BMP | BitMaP |
| CNN | Convolutional Neural Network |
| CR | Character Recognition |
| CRISP-DM | CRoss Industry Standard Process for Data Mining |
| DICOM | Digital Imaging and COmmunications in Medicine |
| DCT | Discrete Cosine Transformation |
| DL | Deep Learning |
| FITS | Flexible Image Transport System |
| GIF | Graphics Interchange Format |
| HOG | Histogram of Oriented Gradients |
| HMM | Hidden Markov Modeling |
| HSV | Hue Saturation Value |
| IP | Image Processing |
| JPEG | Joint Photographers Expert Group |
| KDD | Knowledge Discovery in Databases |
| KNN | K-Nearest Neighbor |
| ML | Machine Learning |
| MNIST | Modified National Institute of Standards and Technology |
| NIH | National Institutes of Health |
| NIST | National Institute of Standards and Technology |
| OCR | Optical Character Recognition |
| PCHR | Personal Child Health Record |
| PDF | Portable Document Format |
| PHOC | Pyramid Histogram Of Characters |
| PHOG | Pyramid Histogram of Oriented Gradient |
| SIFT | Scale-Invariant Feature Transform |
| SVM | Support Vector Machine |
| TIFF | Tagged Image File Format |

# Chapter 1

# Introduction

This chapter is an introduction to the concept of what children's and youth health newsletters are, their current context in medicine, and the medical impact they cause in various areas of study. Also presented is the work to be developed in this dissertation to assist in the process of converting information in paper format to digital format.

## 1.1 Context

Recent developments in the technological world have led to new tools for different areas, such as medicine.

In Portugal there is a valuable collection of information on the evolution of children from birth to adolescence, called Personal Child Health Records, also known as PCHR. Weight, age, height and cephalic perimeter are the values that are registered by pediatricians on a regularly base. Information on the gestation and neonatal period is also recorded.

All the information present in these documents in paper format are important to carry out studies in several areas such as pediatrics, nutrition, population studies, among others.

Through the use of the current processing capacity of the various devices available and low cost, it is possible to carry out processes that were previously time-consuming and costly.

Using techniques in the area of Image Processing (IP), Optical Character Recognition (OCR) and Machine Learning (ML) it is possible to automate processes that were previously performed manually.

The work to be developed throughout this dissertation concerns the use of these techniques in order to provide a platform that automatically performs the process of extracting the information contained in the PCHR, allowing the development of databases with this knowledge that is valuable for medical studies.

## 1.2 Problem description

Pediatric books provide large amounts of information important for several areas. However, most of the information is still in paper format. This makes it difficult to conduct studies in which it is necessary to compare various data relating to several children, making it necessary to convert this information to digital format.

The whole process for manually converting books into information available on a digital platform is quite time-consuming.

Automating the whole extraction process is complex since much of the information is handwritten and in free text, and there is also information to be extracted from graphics.

The handwritten information to be extracted has essentially two problems, one part of free text with annotations relating to gestation and the neonatal period and a part of text, essentially digits, with several values presented in tables.

## 1.3 Motivation

With the main focus of this work being the extraction and conversion of information present in children's and juvenile health books, it becomes very beneficial for the medical area, especially Pediatrics, to make this information available digitally. This makes it easier and quicker to perform medical or population studies.

It will also be interesting to use statistical algorithms or Data Mining to perform automatic analysis using the extracted information.

## 1.4 Goals

The objective of this dissertation is to develop a platform capable of automating the whole process of extracting and converting information present in paper format on PCHRs to digital format. The main expected points to achieve with this project are:

- Automate the entire process of extracting and converting information contained in child and youth health bulletins;
- Facilitate access to the information contained in the bulletins;
- Provide a platform that allows a quick and accurate analysis of data relating to one or more children;
- Allow an increase in quality in the investigation of the Pediatrics area.

## 1.5 Structure of the document

In addition to the introduction, this dissertation contains more 4 chapters. In Chapter 2, the state of the art is described and related works are presented. In Chapter 3 we present the methodology

used, the design of the application and how was the solution implemented. In Chapter 4 a case study is described, where the preparation of this study and the results obtained are analyzed. In Chapter 5, the conclusions obtained with the dissertation are presented, as well as the future work that can be done to improve the final solution.

# Chapter 2

# Image Processing and Data Mining

In this chapter the state of art on IP is described and related works are presented to show what exists in the domains addressed in this dissertation. We also introduce the basic concepts of DM and Text Mining. The main domains are digital image processing, character recognition, text mining and analysis, data mining, and machine learning. The main objective of the application is the collection of information present in individual health records for children and youth through the recognition and extraction of the information contained in the images provided. In Section 2.1, an overview of the image processing is given, containing the various techniques used on the analysis and treatment of images and a comparison of available software libraries. Optical Character Recognition (OCR) is used to interpret and convert the printed or handwritten text to machine-encoded text and is briefly addressed in Section 2.2. In Section 2.4 we present the concept of Text Mining and available techniques used in it.

## 2.1 Digital Image Processing

A digital image is a two-dimensional representation of an image, described as a finite set of digital values, also known as picture elements or pixels [Bha14]. In other words, an image can be defined as two-dimensional function, $f(x,y)$, where $x$ and $y$ correspond to the coordinates, and the value of $f$ is called the intensity or gray level of the image at that point [GW02]. Digital Image Processing (IP) can be very useful in different areas since it provides a way of retrieve lots of information contained on images. According to [ZA15], IP can be divided into three levels as shown in Fig. 2.1 taken from article [SMS12]:

**Low-level -** Primitive operations on the pixel level, like noise reduction, geometric corrections, histogram equalizations, etc.

**Mid-level -** Image analyses, like segmentation tasks and/or classification of individual objects.

**High-level -** Image understanding is high-level operation which studies the relation of individual objects with the environment and/or with another objects presented in the image.

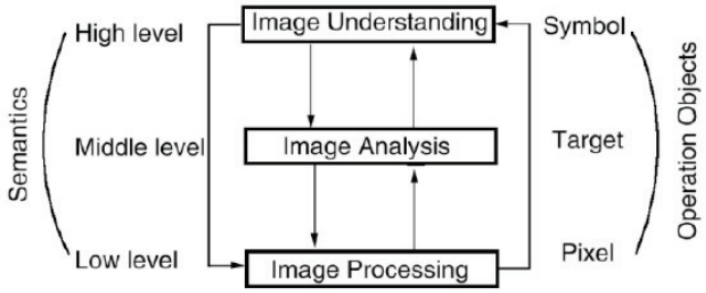



Figure 2.1: Image Processing Levels

In each of these levels, there are categories or steps that must be taken in order to improve the quality of process and analysis of each image. On the next sub-sections are presented the most important ones.

### 2.1.1 Low-level categories

The first step on image processing (**Low-level**) follows three stages, Reconstruction, Transformation and Classification, in which one is subdivided into different categories like is described in [Bha14]:

1. **Reconstruction/Correction**

   **Restoration:** Removal or minimization of image degradations, improving the condition for further processing. There are two types of restoration: radiometric and geometric;

   **Reconstruction:** Projections are used for special class of image restoration problems where a two or higher dimensional object is reconstructed from several one-dimensional projections;

   **Mosaic:** When images are captured in patches, combining two or more to form a single large image without radiometric imbalance. Required to get the view of the entire area.

2. **Transformation**

   **Contrast stretching:** It is used when images are homogeneous i.e., there is no much change in their levels. When analyzing the histogram representation, they are characterized as very narrow peaks;

**Noise filtering:** Used to filter the unnecessary information from an image and to remove various types of noises from the images. Various filters like low pass, high pass, mean, median etc. are available;

**Histogram modification:** Histogram is important in image enhancement, as it reflects the characteristics of an image. Modifying the histogram can directly influence an image, changing it characteristics, e.g., Histogram Equalization;

**Data compression:** Compression is used in order to reduce image size without affecting radiometric properties. Typically it's done by DCT (Discrete Cosine Transformation) developed by JPEG (Joint Photographers Expert Group);

**Rotation:** The images are rotated in order to match with the second image. It is mostly used in mosaic restoration for joining many images together for final interpretation. Most common technique is 3-pass shear rotation;

3. **Classification**

**Segmentation:** Process that subdivides an image into its constituent parts or objects. The level to which this subdivision is carried out depends on the problem being solved, i.e., the segmentation should stop when the objects of interest in an application have been isolated. Image thresholding techniques are used for image segmentation.

**Classification:** Labeling of a pixel or a group of pixels based on its gray value. Classification is one of the most often used methods of information extraction. In Classification, usually multiple features are used for a set of pixels i.e., many images of a particular object are needed to train a classifier.

### 2.1.2 Mid-level categories

According to the work that will be carried out throughout the dissertation, there are two types of image features that have to be segmented and extracted: being manuscript, and printed text and graphs. The reason why character extraction is challenging is because most of the text is written in a table, where in some cases there is an overlap of the text with the borders of each cell in the table.

Since there is a wide range of segmentation techniques depending on the area in which it fits, the most relevant techniques for extracting text will be discussed below.

#### 2.1.2.1 Graph Segmentation

The graphs from which information needs to be retrieved and analyzed are mostly graphs with two variables, such as weight and age of the child (Figure 2.3). To extract information from these graphs it is necessary to resort to more traditional and general techniques for segmentation of images. These can be divided into two main types, Region Based Methods and Edge Based Methods (as shown on Figure 2.2), and there are hybrid methods that combine characteristics of both [ZA15]:

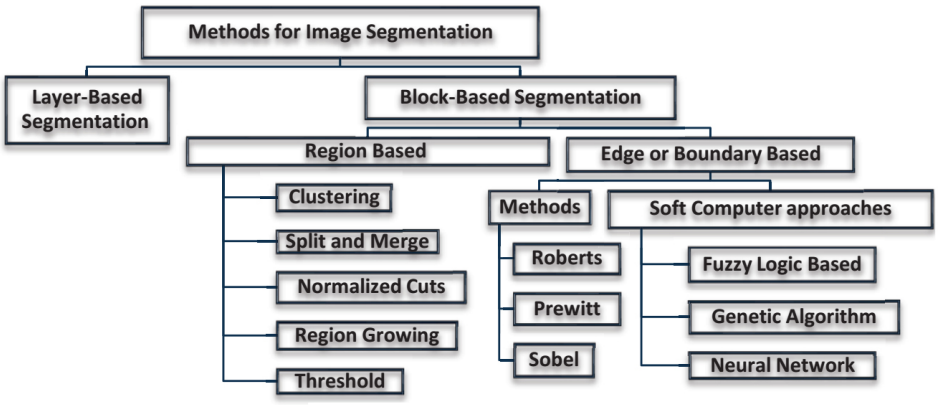



Figure 2.2: Image segmentation methods. [ZA15]

- **Region Based Methods**

  Region Based methods take into account the continuity present in the image, that is, they divide the image into subregions depending on some rules. For example, by taking into account the gray level it is possible to delimit a region whose pixels all have the same value. The main objective is to define regions in the image according to their anatomical or functional functions [SMS12].

- **Edge Based Methods**

  These methods are based on discontinuity, finding abrupt changes in intensity value. They are usually used to detect discontinuities in grayscale images, thus detecting boundaries or edges present in the image. These edges are detected by local changes in image intensity, usually delimiting two regions of the image. Corners, lines, curves are the main features drawn through these techniques [SMS12].

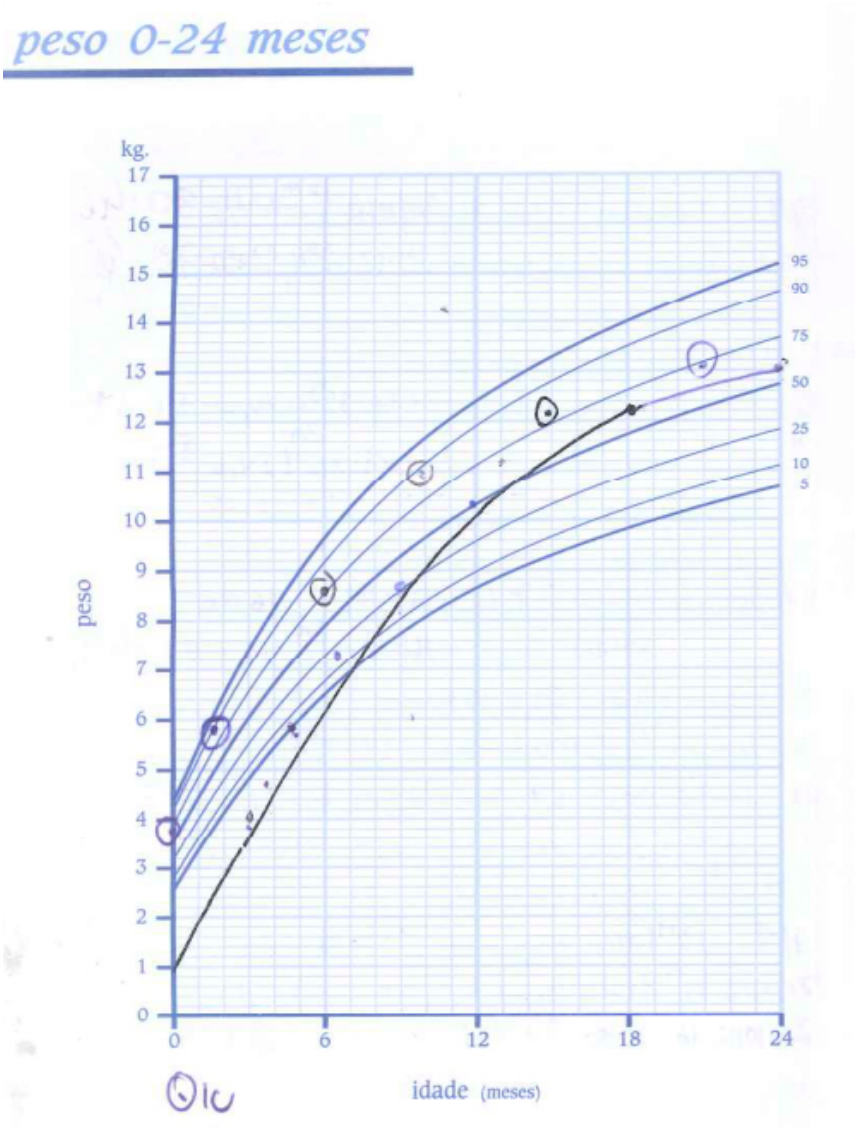



Figure 2.3: Graph with weight and age of the child.

#### 2.1.2.2 Text Segmentation

Based on [MAN+14], there are three levels to achieve text image segmentation, line segmentation, word segmentation and character segmentation:

**Line segmentation (First level)** - It is an mportant step for text based image segmentation. It essentially serves to segment the text line by line, resorting to a horizontal scan of the image, pixel-row by pixel-row from left to right and top to bottom. White space between the text lines is used to segment the text lines [TK12]. As all text may be written with a slight slope or deformation, there must be methods to decrease or correct this problem, in order to improve later implementation of algorithms. Methods are based in using the

projected profile of the image, a form of nearest neighbours clustering, cross correlation method between the lines, and using the Hough transform [MAN+14].

**Word segmentation (Second level)** - Consists of word segmentation, in which the words are segmented by the space between each word [TK12]. Similar to what happens with the first level of segmentation, line segmentation, it is done by vertical scanning of the image, pixel-row by pixel-row from left to right and top to bottom [MAN+14].

**Character segmentation (Third level)** - The final level of text based image segmentation is the character segmentation, in which the objective is to achieve an image decomposed into single characters. It is a critical step for most OCR systems, although is a common cause of errors on these systems. The overlapped text uses projection profile algorithm, connected components and also uses nearest neighborhood method to cluster the connected components [TK12].

#### 2.1.2.3 Segmentation Methodologies

We now present the most common used methodologies in text segmentation. The following list, based on article [MAN+14], lists the various methodologies:

1. **Pixel counting approach**

   On this approach, the line separation procedure consists of scanning the image row by row. The row in the preceding line represents the pixel row and not the lines of the address, i.e. the entire image is scanned from left to right and top to bottom. Then the intensity of the pixel is tested for 0 or 1 (it is considered that is a binarized image). In a binarized image, 0 represents black and 1 represents white. The algorithm would vary according to the image under consideration. Pixel counting approach is a simple technique to implement, but it cannot be used in situations when the text line in the document has a higher degree of skew, when the characters overlap, or when there is irregular spacing between the text lines. There are two ways to achieve line segmentation, one where the document has guidelines and one where the document has no guidelines. In the first case, the separation is done by setting a threshold value for the number of white pixels rows between two lines. In the second case, the presence of the document guidelines is used to delimit the height of each line and thus identify each one.

   Higher level of segmentation can be achieved by minimizing changes in the algorithm logic. For Line segmentation, we perform horizontal cuts along the image length, for word and character segmentation, we have to perform vertical cuts along the width of the image.

2. **Histogram approach**

   Histogram approach is a method to automatically identify and segment the text line regions of a handwritten document. Histogram method can very easily be extended to higher levels of segmentation. A $Y$ histogram is used to segment the text lines, and an X histogram is

used to segment words and characters. An $X$ histogram projection that is applied to each line detected takes out possible words. The points obtained are similar to those obtained from line segmentation. Each cut point reflects a rectangular region where the possibility of a text word/character is maximized. Using this rectangular coordinates, we can extract the words/characters from the digitized image.

(a) **Y Histogram Projection**

The idea is to use a simple and fast method to correctly distinguish possible line segments in the handwritten text. The main objective is that each text line corresponds to a peak in the histogram. The histogram represents the added pixels for each $y$ value. So the empty spaces between the peaks represent possible regions between different text lines.

(b) **Text Line Separation**

Once all the potential lines are detected, a procedure to apply a threshold is performed to obtain a possible line separation in the text. This threshold is dynamically calculated and it is proportional to the average length of the lines in the text ($Y$ histogram values). This procedure aims to remove the regions in the histogram that do not refer to the lines in the text, or the elimination of noises that confuses with the text lines. The choice of the parameter to be used as a threshold is intrinsic and is related to the information like the text. Such an approach restricts the algorithm, thereby utilizing minimum possible of heuristic techniques to determine the line separation points. Actually, this stage tries to identify the location of each text line. The separation of the possible text line regions using the histogram shows a deficit due to the upper and lower regions of some letters

(c) **False Line Exclusion**

This method attempts to remove noise next to text lines regions. Once the possible text line regions are separated by removing an offset from the histogram, it is determined the average height of these regions to exclude false lines that might be detected. If the presence of noise is more than this region poses enough height it can be confused with a text line segment by the algorithm. The height of a line is obtained by taking the limit values of the corresponding region in the $Y$ histogram and calculating the difference obtained by taking the limit values of the corresponding region in the Y histogram and calculating the difference between them.

(d) **Line Region Recovery This**

This method determines the average point between the regions found. The idea is to find the maximum area that each line might be inscribed, by determining the superior and inferior coordinates in the $y$ axis.

3. **Smearing Approach**

   In this method the consecutive black pixels along the horizontal direction are smeared consequently; the white space between the black pixels is filled with black pixels. It is valid only if their distance is within a predefined threshold. This way, enlarged areas of black pixels around text are formed. It is so-called boundary growing areas. These areas of the smeared image enclose separated text lines. Thus, obtained areas are mandatory for text line segmentation.

4. **Stochastic Approach**

   Stochastic method is based on probabilistic algorithm, which accomplished nonlinear paths between overlapping text lines. These lines are extracted through Hidden Markov Modeling (HMM). This way, the image is divided into little cells. Each one of them correspond to the state of the HMM. The best segmentation paths are searched from left to right. In the case of touching components, the path of highest probability will cross the touching component at points with as less black pixels as possible. However, the method may fail in the case that contact point contains a lot of black pixels.

5. **Water Flow Approach**

   The water flow algorithm assumes hypothetical water flows under a few angles of the document image from left to right and top to bottom. In this hypothetically assumed situation, water is flowing across the image. Areas that are not wetted form unwetted ones. The stripes of unwetted areas are labeled for the extraction of text lines. Further, this hypothetical water flow is expected to fill up the gaps between consecutive text lines. Hence, unwetted areas left on the image indicates the text lines. Once the labeling is completed, the image is divided into two different types of stripes. First one contains text lines, the other one contains line spacing. The angle of the flow of the hypothetical water can be obtained using a mathematical function depending on the application.

### 2.1.3 High-level categories

As a final step in image engineering there is an understanding of the image, where the focus is essentially on relating and interpreting the information that has been segmented and extracted from the images with the scene that surrounds them. Since the main focus of this dissertation addresses Character Recognition (CR), according to [JDM00] and [JJKS16] there are four general approaches of Pattern Recognition:

- Template Matching,
- Statistical Techniques,
- Structural Techniques,
- Neural Networks.

These approaches will be further explained and analyzed in section 2.2, since they're are directly related with some methodologies used in OCR.

### 2.1.4 Available Tools

Image processing is used in different contexts, from image correction to analysis of the information contained in each image, in various scientific areas. As such, since there are several algorithms that have been created for a given context and can be used in another, there is a need to facilitate the implementation of all these algorithms. For this were there was a need for creating libraries and frameworks, which having these algorithms already implemented, make their use easier and intuitive. Some of the existing libraries are listed below.

#### 2.1.4.1 AForge

AForge.NET is an open source framework, built in C# and designed for developers and researchers in various areas of Computer Vision and Artificial Intelligence. Implements several libraries, such as image processing, neural networks, genetic algorithms, etc. The framework comes with some application samples, allowing to demonstrate its use. However, it is a project that no longer receives many updates and at the writing date of this dissertation the last update occurred in July 2013.

#### 2.1.4.2 ImageJ

ImageJ[1] is a public domain library for image processing built in Java. It was inspired on an existent image processing and analysis program for Macintosh, called National Institutes of Health (NIH) Image. It was designed with the idea of further extensibility through Java plugins, encouragin users to develop their own plugins, making possible to solve almost any image processing or analysis problem. It's a library that supports many image formats including TIFF, GIF, JPEG, BMP, DICOM, FITS and "raw". It can run on any computer with a Java 1.4 or later virtual machine. It is available for Windows, Mac OS, Mac OS X and Linux.

#### 2.1.4.3 OpenCV

Open Source Computer Vision Library, also know as OpenCV[2], is an open source library that provides more that 2500 optimized algorithms for image processing and machine learning. It was written in C/C++ and it was oficially launched in 1999, by Intel [3]. It supports Windows, Linux, Android and Mac OS, providing interfaces on C++, C, Python, Java and MATLAB. According to OpenCV there is more than 47 thousand people on their user community and they estimate more than 14 million number of downloads.

[1] https://imagej.nih.gov/ij/
[2] http://opencv.org/
[3] http://www.intel.com/

Table 2.1: Comparison between image processing tools

| Tool | Written in | Open source | Cross platform | Image Processing Algorithms | Machine Learning Algorithms |
|---|---|---|---|---|---|
| AForge.NET | C# | ✓ | ✓ | ✓ | ✓ |
| ImageJ | Java | ✓ | ✓ | ✓ | ✗ |
| OpenCV | C/C++ | ✓ | ✓ | ✓ | ✓ |
| Emgu CV | C# | ✓ | ✓ | ✓ | ✓ |

#### 2.1.4.4 Emgu CV

Emgu CV[4] is a cross platform .NET wrapper to OpenCV. This library is fully written in C# and allows OpenCV functions to be called from .NET compatible languages like C#, Visual Basic (VB) and IronPython. It can run on Windows, Linux, Mac OS X, iOS, Android and Windows Phone.

## 2.2 Optical Character Recognition

In recent years there has been growing use of OCR, mainly for two reasons, the increased processing power of devices that are available and the increased use of these techniques in various areas, such as license plate recognition and written document conversion [PMR+16].

OCR is a process that detects and recognizes printed or handwritten characters from an input image and converts it into a digital format [RSCP16]. There are two types of character recognition approaches, offline (a) and online (b), like it can be seen in the Figure 2.4 extracted from the article [RSCP16].

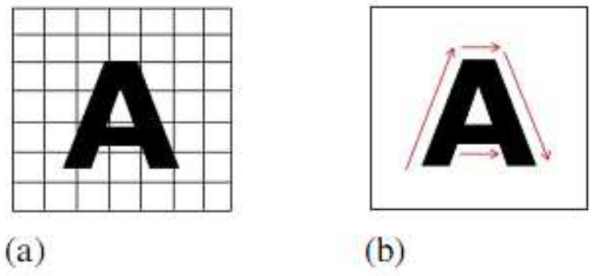



Figure 2.4: (a) Offline character recognition, (b) Online character recognition.

[4]http://www.emgu.com

In the offline character recognition approach the data used in character analysis and extraction are present in scanned images. Whereas, in the online approach the data are represented by a time and order function of the stroke of a specialized pen on a digital surface [PMR+16].

As the data provided for the development of the application during this dissertation will be scanned images, it's on the offline character recognition approach that the focus will be on. On the provided data, there are three types of text that will be extracted using OCR techniques, printed text (a), free handwriting text (b) and digits (c) in a table as it can be seen on Figure 2.5.

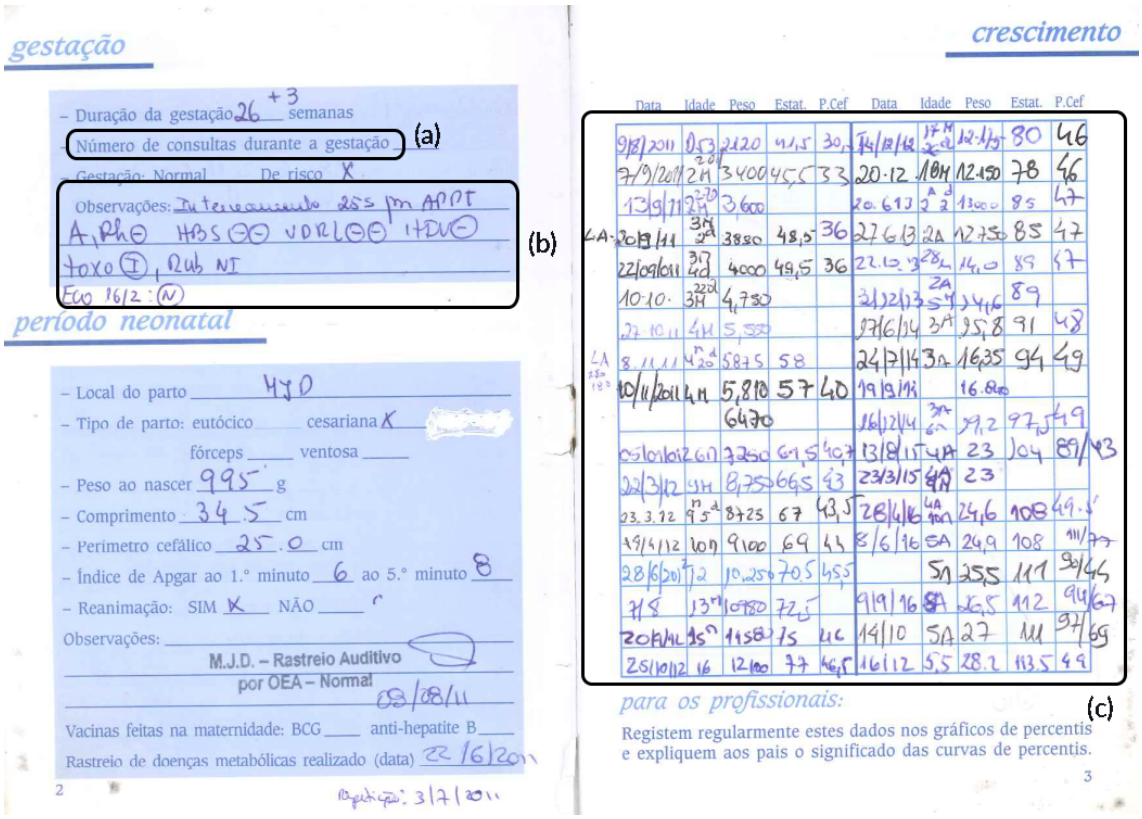



Figure 2.5: An example of a scanned page of an individual health booklet. (a) Printed text, (b) Free handwriting text, (c) Digits in a table.

## 2.3 Data Mining

We currently live in a world where large amounts of information are collected daily. Analyzing all these data becomes an important need for knowledge to be acquired.

Many people treat Data Mining (DM) as a synonym for another popular term: Knowledge Discovery from Databases (KDD). There are others who consider DM as an essential process in the discovery of knowledge.

According to [HKP12], the sequential process for the discovery of knowledge follows the following steps:

1. **Data cleaning** - Remove noise and inconsistent data
2. **Data integration** - Where multiple data sources can be combined
3. **Data selection** - Where data relevant to the analysis task are retrieved from the database
4. **Data transformation** - Where data are transformed and consolidated into forms appropriate for mining by performing summary or aggregation operations
5. **Data mining** - Essential process where intelligent methods are applied to extract data patterns
6. **Pattern evaluation** - Identify the truly interesting patterns representing knowledge based on interestingness measures
7. **Knowledge presentation** - Where visualization and knowledge representation techniques are used to prevent mined knowledge to users

Data is preprocessed in different ways between steps 1 and 4, where data is prepared for mining. The data mining step can interact with the user or with a knowledge base. Interesting patterns may be added as new knowledge to existing knowledge base, after being presented to the user.

This model considers Data Mining as a stage of the KDD process, however several areas consider DM as the whole process of knowledge discovery. Citing [HKP12]: *"**Data Mining** is the process of discovering interesting patterns and knowledge from large amounts of data."*

### 2.3.1 CRISP-DM

CRISP-DM (CRoss-Industry Standard Process for Data Mining) is a non-proprietary and freely available data mining model developed by industry leaders. This model is designed to encourage best practices and allows organizations the framework needed to get better and faster results from data mining [She00].

Crisp-DM organizes the data mining process in 6 phases:

1. **Business Understanding** - Understand and determine business objectives and requirements; Define the objectives of data mining; Produce the project plan.
2. **Data Understanding** - Collect the initial data; Describe the data collected; Familiarize and explore the data; Check data quality.
3. **Data Preparation** - Selection and Cleansing of data; Integration and formatting of data to construct a data set.
4. **Modeling** - Selection of the modeling technique; Generation of test design; Creation and assessment of models.

5. **Evaluation** - Evaluation of results; Process review; Determination of next steps.

6. **Deployment** - Plan deployment; Plan monitoring and maintenance; Production of the final report; Review of the project.

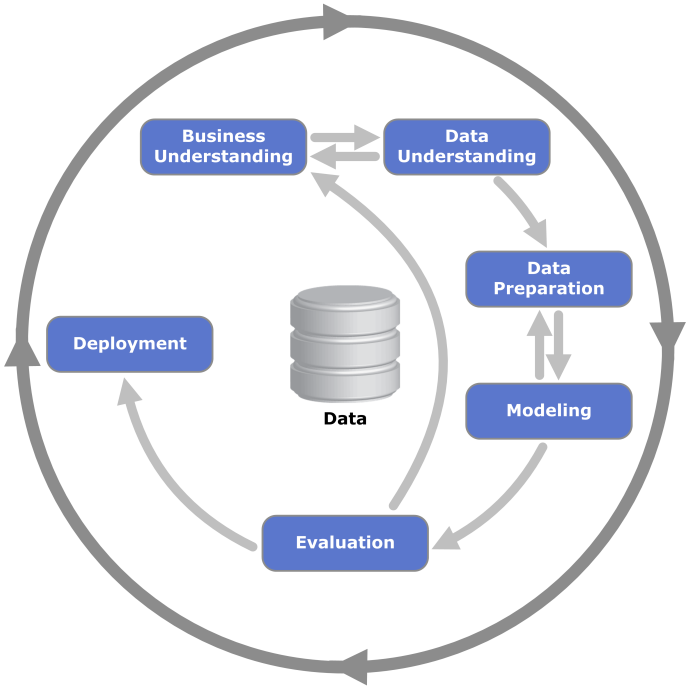



Figure 2.6: Phases of the CRISP-DM model[5]

This data mining methodology allows us to better understand what kind of data is being extracted, as well as what are the main problems to be addressed. This leads to the design of a more consistent and more suitable data extraction model for that type of data.

### 2.3.2 Tasks

Data mining tasks can be classified, mainly, into two classes: descriptive and predictive.

Predictive mining tasks perform induction on existing data to be able to predict. It involves the use of some variables or fields in the dataset to predict future or unknown values of other variables of interest. On the other hand, descriptive mining tasks allow you to characterize the properties of the data in a target data set, presenting interesting patterns that can be used to analyze the data in a different way.

[5]Process diagram showing the relationship between the different phases of CRISP-DM (https://en.wikipedia.org/wiki/Cross-industry_standard_process_for_data_mining) Accessed on 31/01/2018

#### 2.3.2.1 Classification

In this dissertation the most appropriate type of task is the predictive one, due to the fact that it is necessary to implement classification algorithms to recognize the handwritten characters of the tables. As such, it follows the concept of classification in data mining, as well as a brief summary of the most promising classification algorithms, Artificial Neural Network (ANN) and Deep Learning (DL).

According to [HKP12], *"**Classification** is the process of finding a model (or function) that describes and distinguishes data classes or concepts. (...) The model is used to predict the class label of objects for which the the class label is unknown."*.

- **Artificial Neural Network (ANN)**

  Artificial Neural Networks are adaptive nonlinear information processing systems that have as main characteristics self-adaptation, self-organization and real-time learning [DLS+13]. They got their name, neural networks, thanks to the resemblance to the human brain. They are used for various tasks, such as classification, regression, control, modeling and prediction. They are in great expansion due to the constant increase of the use of these techniques by different areas of application [BMD13].

- **Bayes classifiers**

  The most commonly used method for classifying text are the Naive Bayes classifiers [LWJ16]. They are a family of probabilistic classifiers based on applying the Bayes' theorem and are considered simple since the predictive variables are assumed to be conditionally independent given the class [BL14].

- **Support vector machine (SVM) classifier**

  SVMs are supervised learning machines used in two-group classification problems [CV95]. The idea of the machine is to find a hyperplane that separates all the vectors of a class from those of the other class. In addition to performing a linear classification for separable classes, they are also capable of performing a non-linear classification for non-separable classes using the so-called kernel trick.

- **Hidden Markov Model (HMM) classifier**

  An HMM is a probabilistic state machine that follows a Markov process with unknown parameters and one with a set of discrete and finite states, in which the objective is to determine the hidden parameters from the observable parameters. HMM satisfies the Markov chain property in which the next state depends on the current state [AG14].

- **K-Nearest Neighbor (KNN) classifier**

  KNN is a simple non-parametric training method used for classification and regression. In this method, each nearest neighbor is calculated by taking into account the value of $k$, which specifies how many nearest neighbors should be considered to define the class of a sample data point [BA10].

- **Deep learning**

  Deep learning methods allow you to learn Data representations with multiple levels of abstraction. They are obtained through the joining of simple and non-linear modules that transform the representation at one level (starting with the raw input) into a representation at a higher, slightly more abstract level [LBH+15].

## 2.4 Text Mining

Text Mining allows you to automate the extraction of new information, previously unknown, present in different types of written resources through the use of computer algorithms and programs. Text Mining is a branch of Data Mining, which tries to discover and group interesting patterns in large amounts of information [GL09].

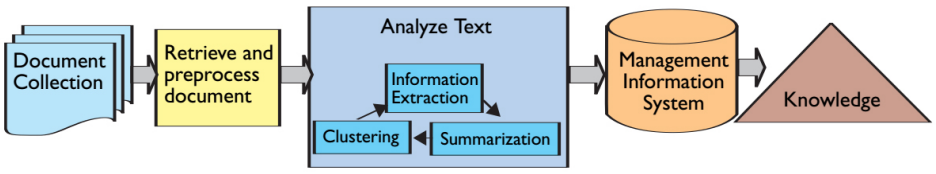



Figure 2.7: Steps on a text mining system. [FWRZ06]

In Figure 2.7 the main steps in a text mining system are described. In the first stage, several documents to be processed in the system are collected. These documents are then preprocessed individually and one or more text analysis techniques are applied. Finally, the collected information is then placed in an information management system to provide knowledge to its users [FWRZ06].

According to [KC16], the available tools for text mining allow several uses, as shown in Figure 2.8.

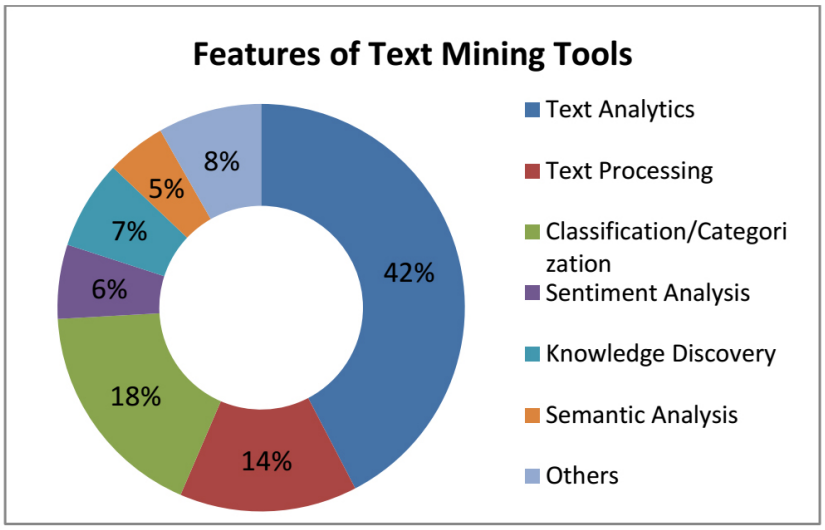



Figure 2.8: Usage of Text Mining tools. [KC16]

### 2.4.1 Text Mining Techniques

In this dissertation, the most relevant step of a text mining system that can help to process and correct problems encountered in the segmentation phase of words present in the image is pre-processing. Some techniques that can be used for pre-processing in text mining are presented in the following list, according to [Gon13] and [Rod16]:

- **Tokenization and removing unwanted characters**

  This process should at the end get a word surrounded by white spaces. To do this, you must remove characters that are unnecessary for the context and then break the stream of the text document into words, phrases or relevant elements, called tokens.

- **Stop Words Removal**

  In text documents there are words that repeat very often, but without essential meaning for the context of the document, because they are used to join words in a sentence. Examples are articles, conjunctions, prepositions, etc. The goal is to remove these words that will not influence the meaning of the document but facilitate later analysis to the text.

- **Stemming**

  It is the process that reduces variations from one word to its common representation, the stem. Examples of this is to reduce plural words, diminutives, among others to their root.

- **Named Entity Recognition**

  Named Entity Recognition allows you to identify and extract terms used to refer to a particular known entity, such as a person, a place, or an organization.

- **Synonyms Handling**

  This technique allows to substitute words present in the document by synonyms, keeping intact the context and meaning of the text. Its main objective is to reduce the number of terms present in the dataset.

- **Word Validation**

  In order to validate the words that have been segmented, this process allows comparing them to words existing in dictionaries. According to the language and context of where the document belongs, different dictionaries will be used.

- **Pruning**

  Pruning intends to eliminate less promising words that may be removed from the document in the initial phase without having a major impact in the context of the document.

## 2.5 Related Work

There are several studies related to the extraction of information present in various types of paper documents. Some of these types are historical documents, digital libraries, bank checks, pre-hospital care report, among other textual sources [RBD+17] [GSGN17].

Depending on the type of document being analyzed, the language in which it is written and whether it is a handwritten or machine-printed document, different techniques are used. According to the survey [GSGN17] performed on techniques for detection of words in documents it can be seen from Figure 2.9 that the number of articles published in recent years on approaches relating to this area of study has increasing.

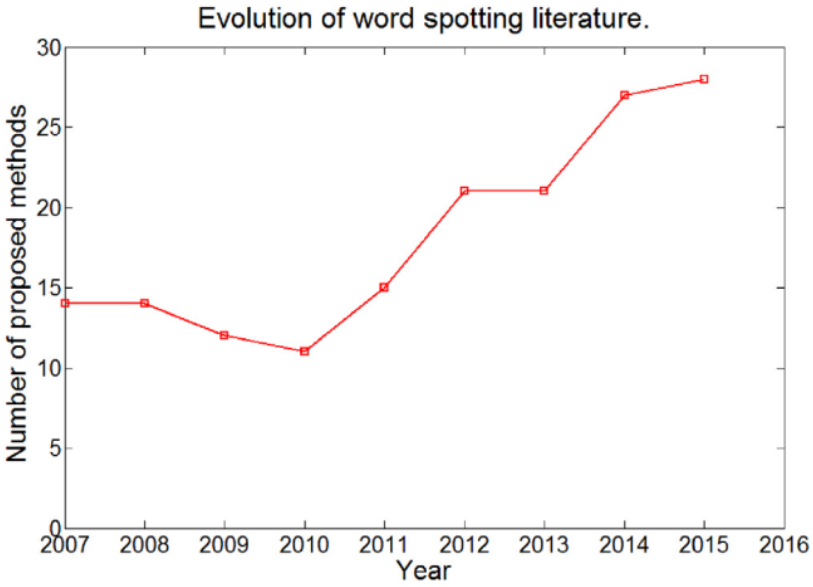



Figure 2.9: Word spotting approaches on the last decade. [GSGN17]

Among the several studies that are presented in this survey and on the current literature, it will be referred to a few studies that are deemed relevant and interesting for this dissertation.

In the Table 2.2 it's presented four studies carried out with regard to the detection and extraction of characters present in handwritten documents, the main focus of the dissertation. According to these studies and the survey [GSGN17] there are several methods for extracting features of the scanned images. The most commonly used features are SIFT descriptors, geometric features and HOG-based descriptors [SF16].

Table 2.2: Recent studies

| Article | Year | Feature Extraction Methods | Representation Methods | Area of application |
|---|---|---|---|---|
| **[RBD+17]** | 2017 | Pyramid Histogram of Oriented Gradient (PHOG) + Tandem | HMM | Health |
| **[RLSG16]** | 2016 | Projections of Oriented Gradients Variants | ✗ | Handwritten Documents |
| **[SF16]** | 2016 | Pyramidal Histogram of Characters (PHOC) | Convolutional Neural Network (CNN) | Handwritten Documents |
| **[AG14]** | 2014 | Discrete Cosine Transform (DCT) | HMM | Handwritten Digit Recognition |

According to the articles presented in Table 2.2 and other studies on the field of handwriting recognition, the most promising recognition methods are HMM and neural networks

## 2.6 Chapter Conclusions

This project will portray several areas, such as image processing, recognition and detection of written characters and machine learning and data mining techniques.

The image processing techniques and methods presented in 2.1 will be useful for a first phase of treatment of the scanned images of the child and juvenile health bulletins, segmentation of the important parts and classification of the information segmented. Segmentation and classification will be performed taking into account three different types of tasks, such as handwritten text recognition, handwritten digits recognition and identifying points in two-variable graphs.

The OCR techniques referred to in 2.2 will be useful for the identification and classification of manuscript text, as well as the handwritten digits present. Also referred to in the 2.5 section are several studies carried out in the area of recognition and extraction of handwritten information where it is possible to recognize that the most widely used methods with the most promising results are Hidden Markov Model and Neural Networks.

Text Mining techniques, referred to in 2.4, will be applied to validate and complete words or phrases to be recognized.

# Chapter 3

# Methodology and Implementation

This chapter describes the methodology used throughout the dissertation, the flow of the application pipeline and the steps used to implement it.

## 3.1 Methodology

Throughout this section will be described the methodology used throughout the project, such as datasets and tools used.

### 3.1.1 Application Flow

As mentioned earlier, this Dissertation is a proof of concept to help in converting printed information from child health books to digital format.

In order to achieve this goal, a flowchart has been planned for the application to follow as we can see in Figure 3.1.

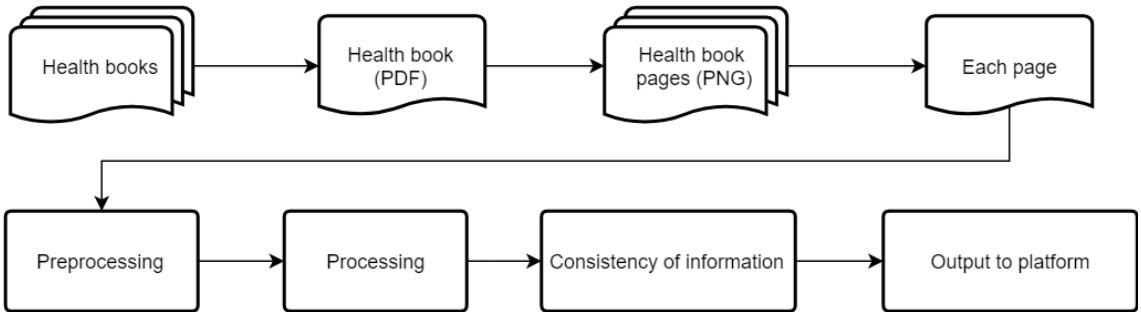



Figure 3.1: Application flowchart

At the beginning of the pipeline, the first step is to separate each .PDF file that represents a book into several individual images, assigning them the name according to the page number. In this step a folder is obtained for each file (book), being named with this one and containing an

image for each page. Then, given the number of the page that is being preprocessed, we can have information about what kind of page it is, whether it is a page with a chart or a page with the table that contains the records made by the doctor.

At this point, the type of page that is passing through the pipeline is known, requiring image preprocessing to remove noises, problems, and to detect certain values to be used in processing. Values such as the range of the predominant color of the image and the region of interest of the image where the table/chart is located.

As we can see on the Figure 3.2, depending on the type of page there are some different steps to approach during the preprocessing. On images that contain charts we focus on finding the axis and in removing the grid, leaving only the points marked by the doctor. On pages with the information written in the table we focus on clearing the image to isolate only the handwritten characters.

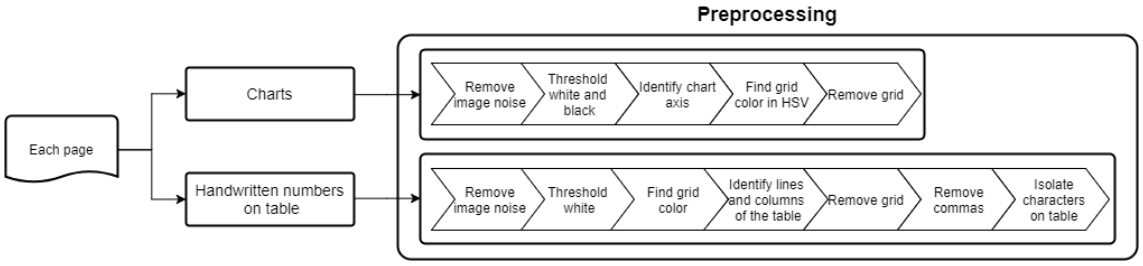



Figure 3.2: Preprocessing flowchart for each page

After isolating the handwritten characters from the table and the points marked in the chart, and also having identified the rows and columns of the table and the axes of the chart, each image must be processed.

In the table it is necessary to divide the processing in two phases, the segmentation and the classification as can be seen in Figure 3.3. The segmentation aims to detect each table cell and then analyze cell to cell to segment characters present in it. However, given the degree of difficulty in splitting characters correctly, several are not well segmented. In the future, it is necessary to create a process that analyzes these cases and separates them correctly. Characters that are well segmented are then normalized and saved as an image. The classification aims to train a classifier that will be used to predict previously segmented characters. For this, three datasets, the MNIST, the manual dataset, and a combination of these two are used.

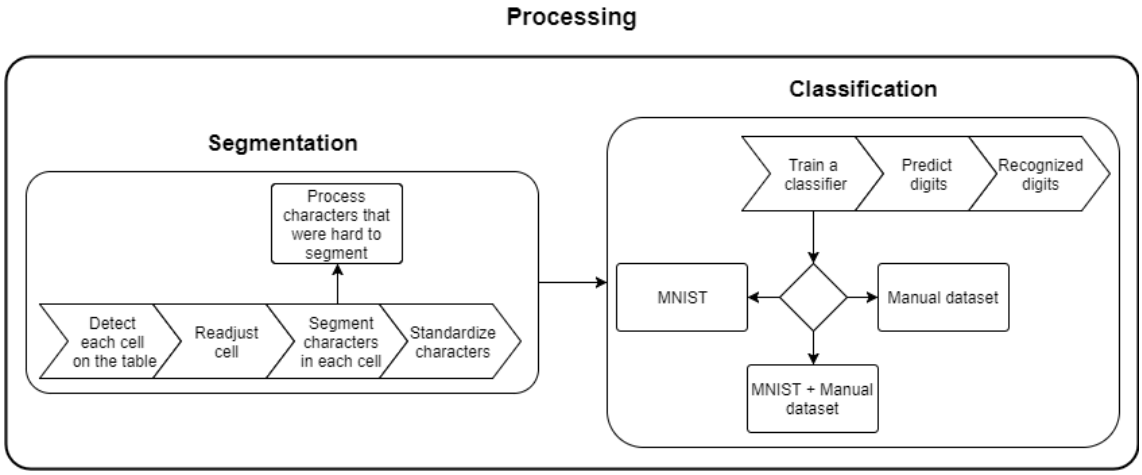



Figure 3.3: Processing flowchart for table

On pages with graphs the processing is performed more easily since the biggest problems have already been pre-processed. In this case, like is shown on the Figure 3.4, the image processing needs to recognize the points marked in the graph and then calculate the values relative to each axis, taking into account what type of chart is involved.

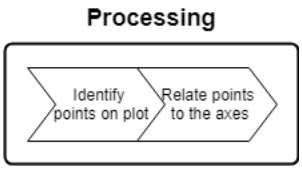



Figure 3.4: Processing flowchart for charts

Finally, as can be seen in the Figure 3.1, in order to validate the information obtained by the processing of the charts and the table it is necessary to test the consistency between them.

The results from the segmentation and classification of the digits may not be sufficient, being possible to have characters badly segmented or that have not been classified efficiently. To solve this problem there is an exchange of information between the identified and unidentified characters of the table with the information extracted from the chart, testing the consistency of both.

In the figure 3.5 are represented in green the characters that are well segmented and classified, whereas the red one is the opposite. Those that were well segmented and classified form a variable that maintains the order of the characters in the cell, as well as the integer values of each one. The bad segmented and classified also maintains its corresponding position in the cell, but are stored as an undefined value.

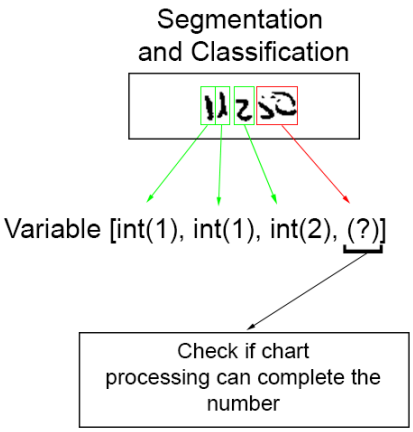



Figure 3.5: Consistency of information diagram

Subsequently, cells with undefined values use the values found in the chart to see if the corresponding value can be found. If it is not possible, this character is stores as undefined and is later questioned to the user in the interface of the platform. With the value set by the user in the interface, the corresponding image and the value set by the user will be stored for future usage.

The platform interface allows the user to view the information that has been extracted from the corresponding table and charts.

## 3.2 Implementation

In order to simplify the problem, the solution implemented was directed to the female child health records. This selection was chosen because it is easier to distinguish the pen colors from the color of the grids.

As previously stated in 3.1, several scripts were developed in Python to create a pipeline to go through each set of books provided. This approach has been adopted to make the project more modular, that is, to make it relatively easy to restructure or improve some phase of the pipeline. As long as the change, improvement or addition of scripts in the pipeline respects the input and output of each phase there should be no complications.

Throughout this section, the main approaches taken in the implementation of the developed code will be described.

### 3.2.1 Page splitting

In a first step it is necessary to divide the PDF files into individual images of each page of the book. Each file follows a similar structure as there are two pages of the book on each page of the PDF. This makes it necessary to initially convert each PDF file page in image format and divide this into two, each page of the book.

For this, it is initially necessary to convert each PDF page to an individual image. Each of these images will correspond to two pages of the book. After loading each image, it is converted to a grayscale to facilitate its treatment. A Gaussian blur is applied to reduce image noise. Then it is necessary to see the orientation in which lies the page, vertically or horizontally. If in a vertical position, an image rotation is performed so that it is horizontal. To split each image into two in order to obtain separate book pages, we begin by applying an adaptive thresholding where the method used to calculate the threshold value is the mean of neighborhood area. With the binary image the histograms will be calculated in order of the two axes, $X$ and $Y$, which will serve to calculate the boundaries of the page, where the number of black pixels is superior to the rest of the image. After knowing the lateral limits of the pages, the central value is calculated by dividing by two the absolute value between the difference of the lateral ends. Starting from this value and using its neighborhood, we will analyze the histogram of the $X$ axis and choose the most appropriate value to divide the page. Since we have all the necessary values to define the region of interest for each page of the book, images are saved for each page.

### 3.2.2 Table page

In the pages that contains the tables with the values recorded by doctors over time it is necessary to face some problems:

- Identify table rows, columns, and cells
- Isolate and segment the characters
- Train a classifier for isolated characters
- Predict the characters isolated with the trained classifier

#### 3.2.2.1 Table cells identification and characters isolation

Initially we load the image referring to the table and apply a Gaussian blur to remove some noise and normalize the colors of the image. Next, a threshold is applied to the image to remove the white from the image. For this, a threshold is applied in BGR in which the colors that are in the three channels (blue, green and red) between the values 230 and 255 are removed, leaving an image similar to that which can be seen in Figure 3.6.

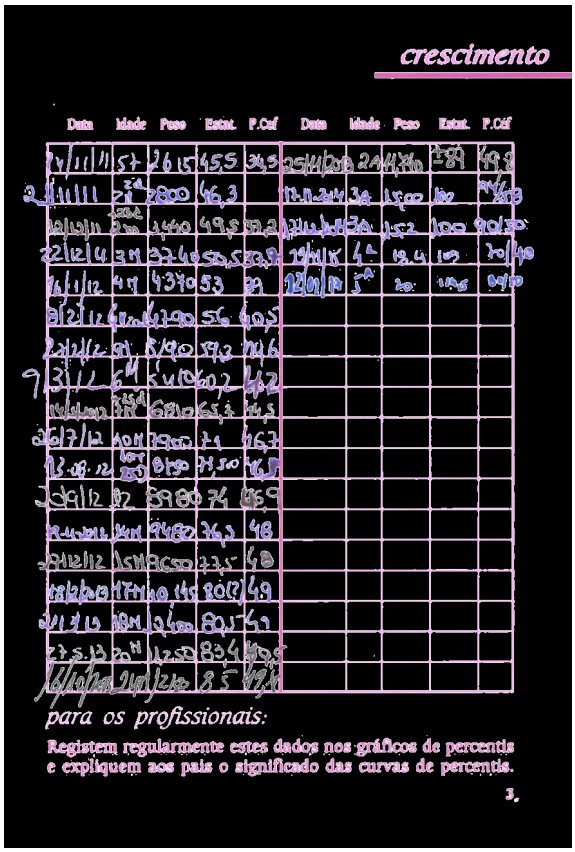
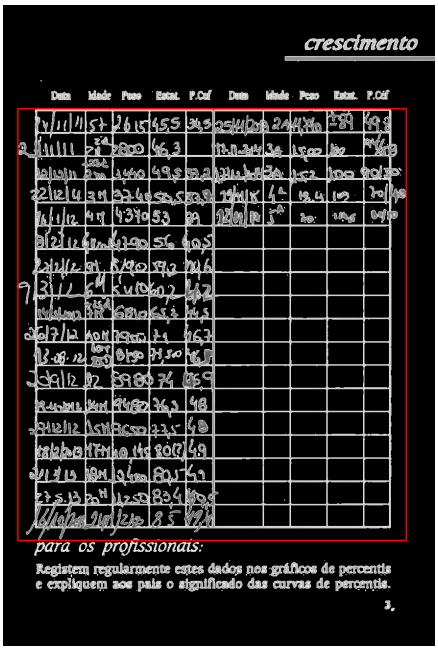


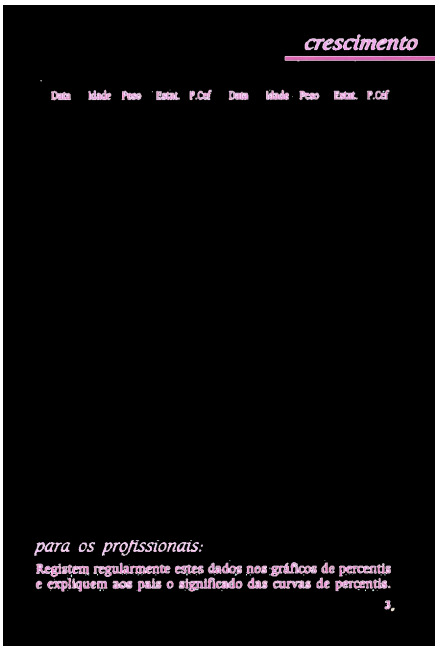

Figure 3.6: Table image after removing white color

Then, the coordinates that delimit the table are detected using an existing OpenCV function called *findContours* that is able to detect contours in an image that is in a greyscale. Therefore, a grayscale image of the page without white is used and to assure that the lines are detected correctly a morphological transformation, closing, is performed with a horizontal kernel, which will close gaps that exist in the horizontal lines. On Figure 3.7a is possible to see a representation of the limits found for that table.

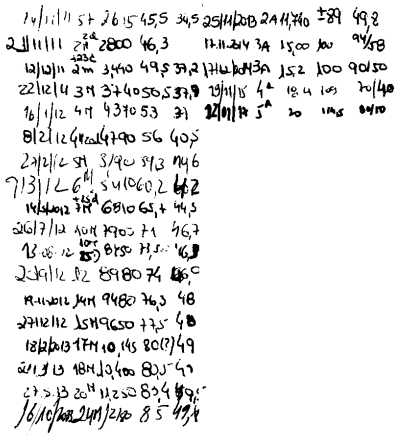
With the coordinates of the table we can create a mask that isolates the part above and below the table, as shown on Figure 3.7b. This mask is then used to calculate the histogram on the three BGR channels and calculate the range of colors in the table grid. Note that this works well when there is no other color except the color of the table grid. In order to reduce the risk of finding some kind of pen strokes or colors outside the table grid, the histograms calculated for the HSV mode image are analyzed and the predominant color is given priority.




(a) Limits of the table




(b) Mask



(c) Characters isolated

Figure 3.7: Steps along the isolation of the characters on the table

After obtaining the color values of the table grid, it is removed by using a threshold between the previously calculated minimum values for each channel and 255. Since there is a vertical center line which is a slightly more intense color, a detection of that same line is done and is removed afterwards. Then, after a few more adjustments and binarizing the image, something similar to the one shown in Figure 3.7c is obtained, where the isolated characters of the grid can be found.


(a) Table without characters

(b) Table with lines and columns detected

Figure 3.8: Steps along the detection of lines and columns of the table

At this point it is necessary to identify the rows, columns and cells in the table. Using the image that was obtained with the recognized characters, one can get the table without the characters, performing a subtraction of images between the initial table and the isolated characters. Then two morphological transformations are performed, one opening vertically and the other horizontal, in order to reduce the number of gaps between the lines that delimit the table, as shown on Figure 3.8a. Finally the histograms are analyzed in the two axes and the values of X and Y are calculated for each outside line. With the bounding lines of the table, to identify the columns use percent values relative to the size of each column. We divide the size of the vertical line that delimits the table by the number of rows of the table to get the height of each row. Once the values of the columns and the rows have been obtained, just iterate over them to get a list of the cells. In Figure 3.8b we can see a representation of the rows and columns detected and it's possible to find some parts where the characters overlap the cell boundaries. To solve this problem it is done a readjustment of the cell boundaries, looking at the neighborhood of each boundary. The histogram for each side of the limit is calculated and if zero black pixels are found next to the limit, the new $X$ or $Y$ is calculated considering this. If this is not possible, the side that has more black pixels is chosen to reduce the risk of losing information.

#### 3.2.2.2 Character Segmentation

To segment the characters of each cell, the x-axis histogram is analyzed and the histogram indices are calculated where the number of black pixels is greater than 2 and are consecutive. Then the aspect ratio of each character or set of characters that respect the previous condition is calculated. Only well-segmented characters are considered if the aspect ratio is less than or equal to 0.6 being stored in a list of well-targeted characters, otherwise they are placed in a list of characters that were not well segmented. In Figure 3.9 you can see the original cell and the characters that were well segmented (3.9b, 3.9c and 3.9d) and those that were not well segmented (3.9e).

(a) Original (b) (c) (d) (e)

Figure 3.9: Steps along the segmentation of characters in a cell

To prepare the characters as input for the classifiers they are saved as image as follows:

- 28x28 pixels, with the character in white and the background black
- the character should be centered in the middle and should touch the top and bottom of the image

To do this we first make a clipping of the image, getting only the pixels of the character and then a scaling is done to be 28 pixels in height or width. If a vertical scaling was done is made a horizontal padding to obtain 28 pixels wide, otherwise it is made a vertical padding to obtain 28 pixels high. After the scaling and padding the character is stored in a specified folder identified as characters well segmented, as shown on Figure 3.10. The characters that were not well segmented are saved as well, but in another folder. To provide information about which cell and where in the cell the character belongs is saved with the following name:

*"(column-number)_(line-number)_(character-number).png"*

i.e. 2_17_2.png (character 2 of the column 2, line 17)

(a) (b) (c)

Figure 3.10: Characters after scaling and padding

#### 3.2.2.3 Classification and prediction

Neural networks were the type of classifier chosen to recognize the digits in the table, which are relatively easy to implement and capable of producing interesting results. The Keras allowed to create two models of neural networks, one conventional and one convolutional:

- Conventional - This model is a simple neural network with two hidden layers with 512 neurons each. The input layer receives a 168 feature vector. A rectifier activation function is used for the neurons in the hidden layers. To avoid the classifier from overfitting there are done 2 *Dropouts* of 20% of the neurons between the hidden layers, preventing the model to co-adapting too much.

  As an output layer it is used a softmax activation that ouputs into probability-like values, allowing one of the 10 classes of digits to be selected as the classifier's output prediction.

- Convolutional - This model is a bit more complex and involves the use of convolutional layers and pooling layers. With the help of Keras examples[1] and a tutorial found in a blog[2] the following network configuration was obtained:

  1. Convolutional layer with 30 feature maps of size 3 x 3.
  2. Pooling layer taking the max over 2 * 2 patches.
  3. Convolutional layer with 15 feature maps of size 3 x 3.
  4. Pooling layer taking the max over 2 * 2 patches.
  5. Dropout layer with a probability of 20%.
  6. Flatten layer.
  7. Fully connected layer with 256 neurons and rectifier activation.
  8. Fully connected layer with 100 neurons and rectifier activation.
  9. Output layer.

The results obtained by each of the models will be discussed in the next chapter, as well as the datasets and features used.

[1] Keras CNN example (https://github.com/keras-team/keras/blob/master/examples/mnist_cnn.py) Accessed on 05/01/2018

[2] Handwritten Digit Recognition using Convolutional Neural Networks in Python with Keras - Machine Learning Mastery (https://machinelearningmastery.com/handwritten-digit-recognition-using-convolutional-neural-networks-python-keras/) Accessed on 05/01/2018

To predict each character a script was created that will iterate over each saved image of the characters that were well targeted. By choosing the model on which to predict, the 3 digits with the highest precision value are selected and placed in a list. Each element of this list will contain the following information:

- Cell line and column
- Image file names of each cell character
- Top 3 digits of prediction for each character
- Percentage of the top 3 digits

The list is finally saved in a file for future processing.

### 3.2.3 Chart page

On the pages that contain charts, it is necessary to address the following issues:

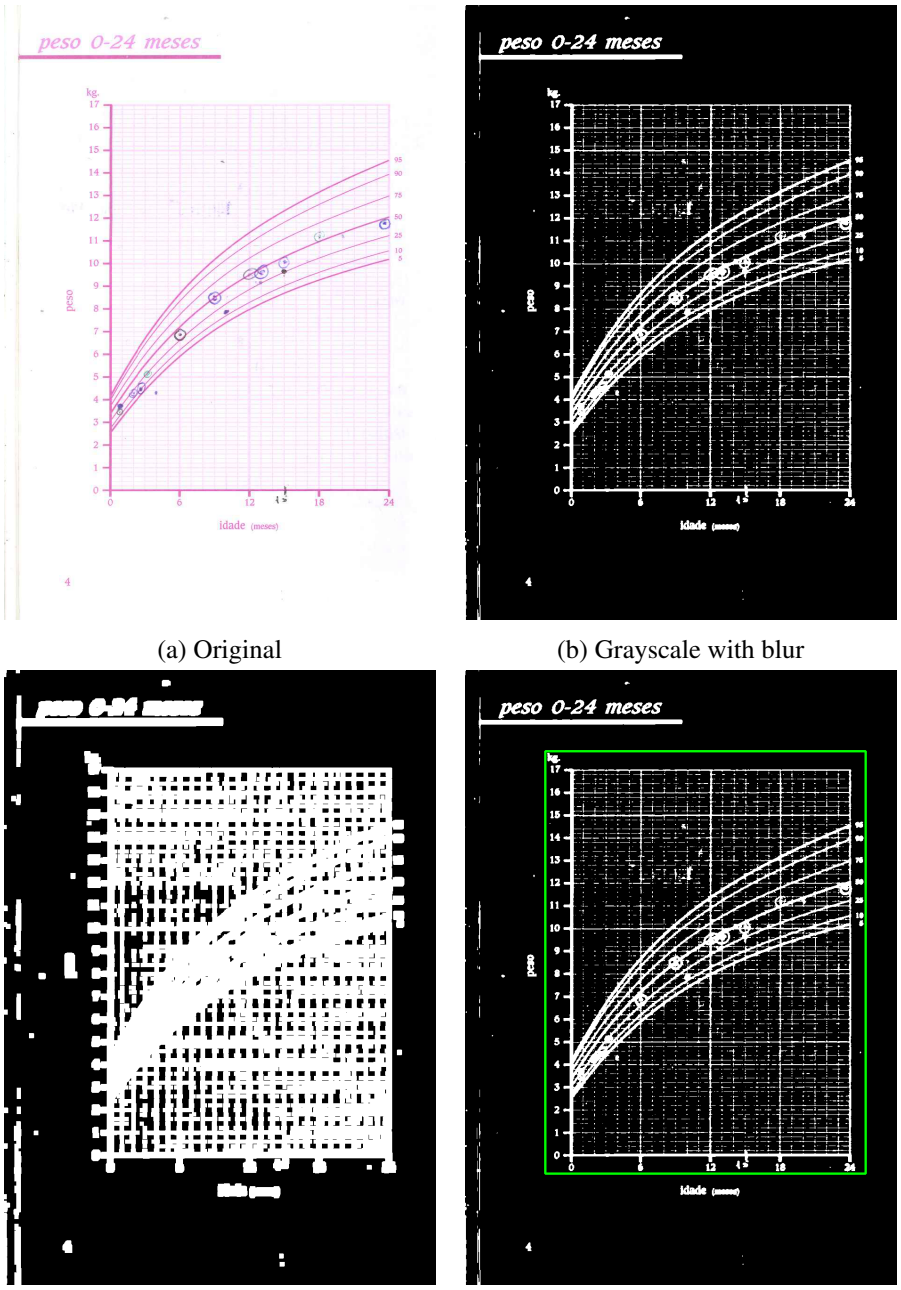
- Detect chart boundaries
- Identify the chart axes
- Isolate and identify the points marked by the doctor
- Relate the points to the axes

#### 3.2.3.1 Chart boundaries

The chart image is loaded in BGR and a Gaussian blur is applied to reduce image noise. A conversion of the image to grayscale color scheme is made. As the goal is to identify the boundaries of the graph, a morphological transformation, dilatation, with a 9x9 kernel is initially done.

Finally, the *OpenCV findContours* function is used to find the respective area of the graphic by identifying the contour whose area is larger than 1/4 the size of the initial image is selected. The bounding rectangle of the contour found is calculated and the values of $X$ and $Y$ where the chart is found are returned. With these values the image is cropped, obtaining only the chart.

It is possible to see the different stages through which the image passes in Figure 3.11.



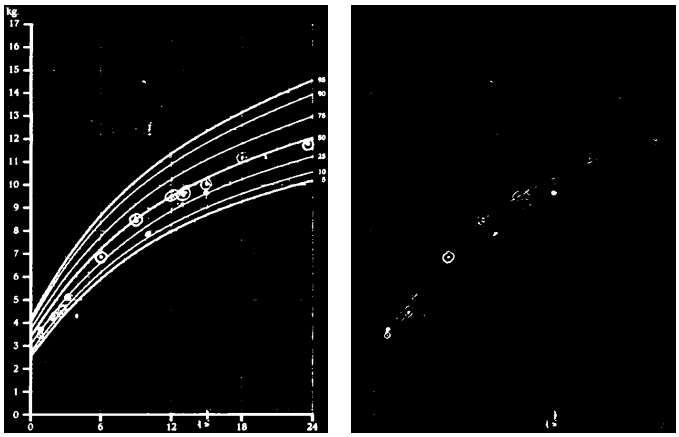

(a) Original (b) Grayscale with blur

(c) Dilation (d) Chart boundaries lines

Figure 3.11: Chart boundaries detection

#### 3.2.3.2 Chart axis identification

Similar to what was done to table pages it is made a mask (Figure 3.14a) to remove the white color of the chart image but is also created a mask (Figure 3.14b) to detect darker ink colors (like black) that are below 190 on all BGR channels.



(a) White mask (b) Black mask

Figure 3.12: Chart masks

The axes will be found by using an *OpenCV* function called *HoughLinesP* that uses the probabilistic Hough line transform. The image used is the mask created to remove the color white from the chart image as it is already in grayscale. This function returns the values of $X$ and $Y$ for the extreme points of lines found. With a minimum line size greater than 200 pixels and with the maximum interval between two points to be considered on the same line being 2 it will be possible to return the axis lines.

Since more than one line is to be detected on each axis, because the axis lines are more than 1 pixel in thickness, the largest vertical and horizontal lines are calculated. Then the value of $X$ is corrected for the horizontal line and the value of $Y$ for the vertical line at the point where the axes intersect. Finally, the extreme values of the two points representing each of the lines of the axis are returned. In Figure 3.13 we can see the lines identified.

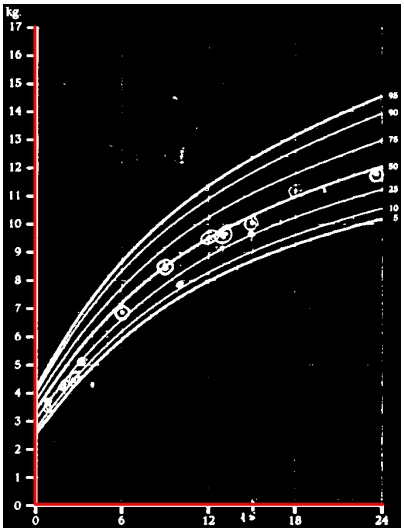
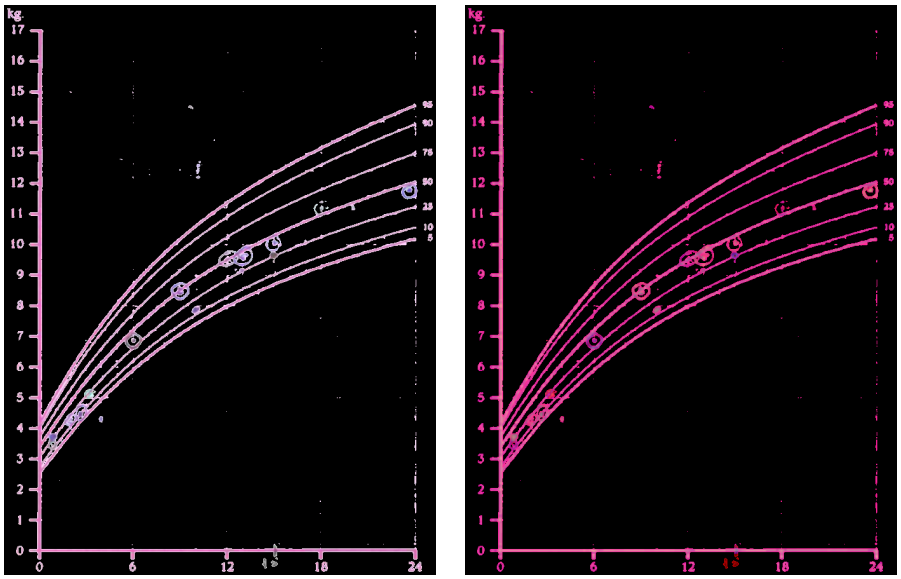


Figure 3.13: Identified chart axes

#### 3.2.3.3 Isolation and identification of the marked points

With the chart image without white (Figure 3.14a), a conversion is made to the HSV color model (Figure 3.14b). This conversion is performed to facilitate identification of the predominant color which is easier to identify through hue. By splitting the HSV image into its three channels, a hue histogram calculation is performed. With this histogram we will calculate the highest value, thus identifying the predominant color.



(a) BGR (b) HSV

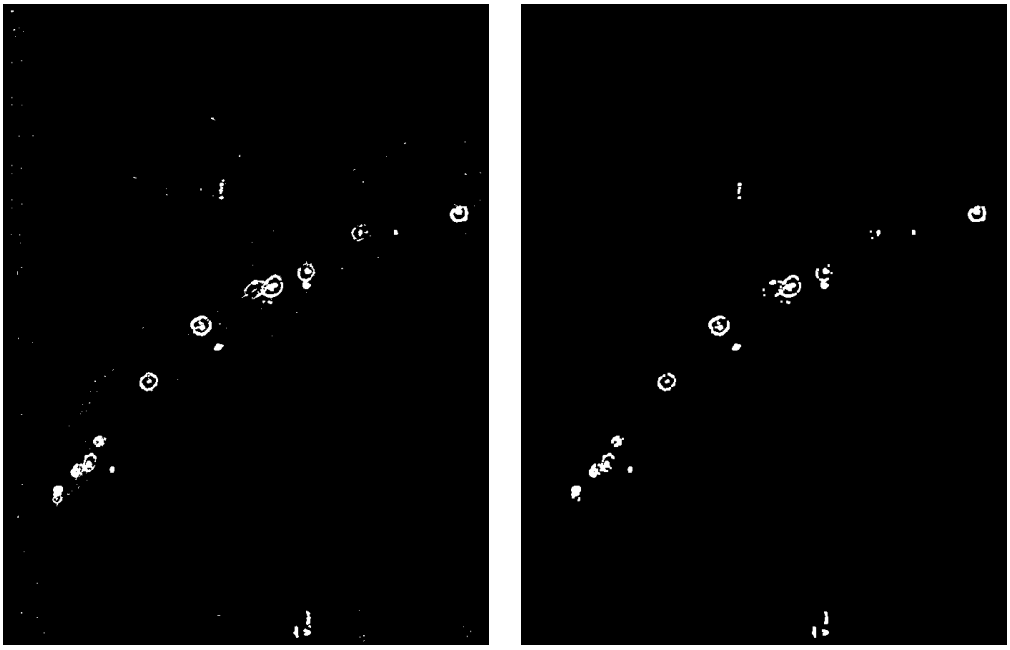
Figure 3.14: Chart image without white

In a normal representation, the hue is represented by a circle with 360º, however, in *OpenCV* is represented only by 180º. Usually the circle is divided into 6 to represent the main tonalities, which in *OpenCV* is equivalent to 30º, that is, after calculating the index with the maximum value in the histogram, the range to define the predominant color of the graph grid in hue goes to be:

$$Range = [maxvalue\_ind - 15, maxvalue\_ind + 15]$$

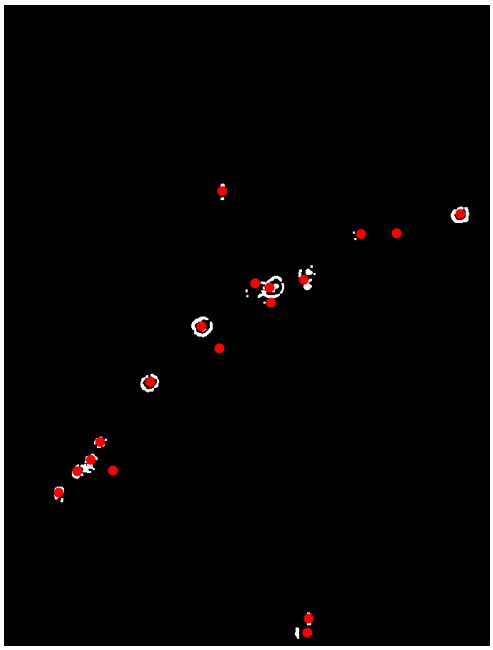
With the range found, the mask is created through a threshold to the hue channel. Since dark ink colors may have been eliminated because they vary in saturation and value channels, the previously calculated dark ink color mask is used and combined with the hue channel mask to obtain a new mask, as shown on Figure 3.15a.

This new mask allows to isolate the points marked by the doctor relatively well. To improve the mask is done an opening (Figure 3.15b), with a 3x3 kernel, that allows to eliminate some noise that still can exist in the image, without causing great impact in the identified points.



(a) Combined mask

(b) After opening



(c) Detected points

Figure 3.15: Chart points detection steps

To check the coordinates of each point, we used the *findContours* function again. Can only be considered as a new point, the contours that meet the following conditions:

- Area bigger than 15 pixels
- Contour is not inside another contour
- The center of the minimum enclosing circle has no point closer than 20 pixels distance

As shown in Figure 3.15c, the contours that meet the conditions are detected as points. Finally we get a list of coordinates with the center of each point detected.

#### 3.2.3.4 Relationship of points with the chart

To correlate the points with the chart is necessary to know what chart is being analysed in order to obtain the axes scales and the units of measurement used. To do this, the file name of the image being loaded is used, allowing us to call the appropriate function to handle it.

With the chart axes and their scales it's possible to mark their divisions on the chart. Then with the abscissa it is possible to calculate the relative value in the $X$ axis and the ordinate value allows to calculate the relative value of the $Y$ axis.

Finally, a list of the values for each point is obtained and exported into a file for future processing.

### 3.2.4 Datasets

To obtain a neural network capable of identifying handwritten digits, it is necessary to use a set of previously labeled digits in order to train and test the performance of the neural network.

Handwritten digits can sometimes be complex in their identification depending on how well or not the image was segmented, as well as the quality, quantity, and diversity of the dataset used to train the neural network.

In the context of this dissertation, two datasets, the MNIST and a cut-out character dataset of the health bulletins, were used.

#### 3.2.4.1 MNIST

MNIST is one of the most commonly used data sets for the purpose of identifying handwriting digits.

It is a subset of some of the NIST (National Institute of Standards and Technology) datasets and is oriented only to handwritten digits. Each digit is represented by a grayscale image of 28x28 pixels. The handwritten digit is centered in the image and is white on a black background.

This dataset has a large dimension totaling 70000 digits, where 60000 are used to train and the remaining 10000 to test.

#### 3.2.4.2 Extracted dataset

Some examples of characters in health bulletins were used to see if it would be easier to identify the correctly segmented characters present in the table with the child's growth information.

These were obtained using an image-editing tool, performing a grayscale conversion and then a threshold to highlight the characters. They were then cut out and stored as a separate image. 600 characters were cut, 500 to train the neural network and 100 to test the same, ie. 60 characters of each digit were cropped out.

### 3.2.5 Tools and Libraries

There are two major areas that are addressed in this Dissertation: image processing and machine learning. Image processing is necessary since the input of data is done through images, it being necessary to use image processing techniques to remove from them the relevant information present in them. The different types of image processing levels were addressed in 2.1. Machine learning is used to train a classifier to identify handwritten digits obtained through the processing of the table image which contains the values recorded by the doctor.

#### 3.2.5.1 OpenCV

Previously described in 2.1.4.3, this library will allow us to run algorithms used for preprocessing and image processing in an easier way. It was decided to choose this library due to its vast amount of IP functions. Version **3.3.0** was used during the course of the dissertation.

#### 3.2.5.2 Keras

Keras is a neural network API written in Python and allows an easy implementation of neural networks, making it capable of creating and modifying neural networks quickly. This ability to easily modify the neural network was the main factor to choose this tool, to allow faster test different types of neural networks. Version **2.0.9** was used.

The Python programming language was chosen due to its compatibility with the two chosen libraries and also because it is one of the languages most used by the DM community.

# Chapter 4

# Case study

Throughout this chapter the restriction conditions for the PCHRs used and the features chosen for the implementation of the classifiers are presented. Some steps of the image processing are also presented and the results obtained for the two types of classifiers are discussed.

## 4.1 Setup

In order to achieve such results, it was necessary to choose books that were considered good test samples and that respected some restrictions made at the time of implementation.

As such, PCHRs were chosen that respected the following conditions:

- Female PCHR
- The table page with records does not have too much information outside the table
- Books that do not have too much use, like tears and folds

Five books have been identified that respect the conditions, however, throughout this chapter it will be demonstrated images of the results obtained for 1 book in specific.

As for the training of the classifiers, 3 datasets were used, which were previously described in 3.2.4. To ensure that the character images used by MNIST were in accordance with how the characters were extracted from the table, for each of the MNIST characters the same function was used to standardize.

In table 4.1, it is possible to see the features used for input of the training of each classifier and dimension of the input vector.

Table 4.1: Features and input size for classifiers

| Model | Features | Input vector dimension |
|---|---|---|
| **CNN** | Image pixels normalized | 784 |
| **NN** | X and Y Histograms<br>Grid<br>Diagonal | 168 |

In the convolutional neural network the normalized values of all the pixels of the image are used, of size 28x28 pixels, which makes a total of 784. In the case of the neural network, the features defined by us are extracted for each one of the images are as follows:

- X and Y Histograms - Two histograms are calculated to count the number of black and white pixels, one on the *X* axis and the other on the *Y* axis. These values are normalized by dividing by 28, which returns a fractional value between 0 and 1.

- Grid - Each digit is divided into one grid and for each cell in that grid the number of white pixels is calculated. This number is normalized by taking into account the size of the cell, returning a value between 0 and 1. Each image of a character was divided into a grid of 7x7, making a total of 49 cells. In the image 4.1 you can see a representation of this for an image with the digit 2.

- Diagonal - This feature extraction divides the image into equal areas and features are extracted in each zone along its diagonal [PSH11]. This feature extraction was one of the best results for neural networks, according to [Pat14]. For our case each digit was divided into 49 zones and each zone had the dimension of 4x4 pixels.

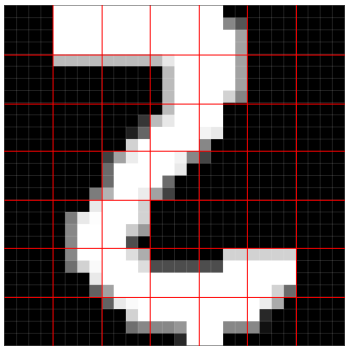


Figure 4.1: Character grid feature

## 4.2 Results

### 4.2.1 Image processing

In the section 3.2 the main steps regarding the pre-processing and processing of the images of the tables and charts have already been described.

The results obtained for the table are in Figure 4.2 and for the chart in Figure 4.3.

In the table, well-segmented characters are identified by a green rectangle, however, it is possible to see that there are cases where the characters are not well isolated, sometimes having two very close characters that are segmented as one. In these cases, the value assigned by the classifier is not correct. Characters that have been poorly segmented and are identified as such are marked by a red rectangle.


Figure 4.2: Characters segmented from the table image

Overall, for the table in question most of the important digits in the table have been segmented successfully.

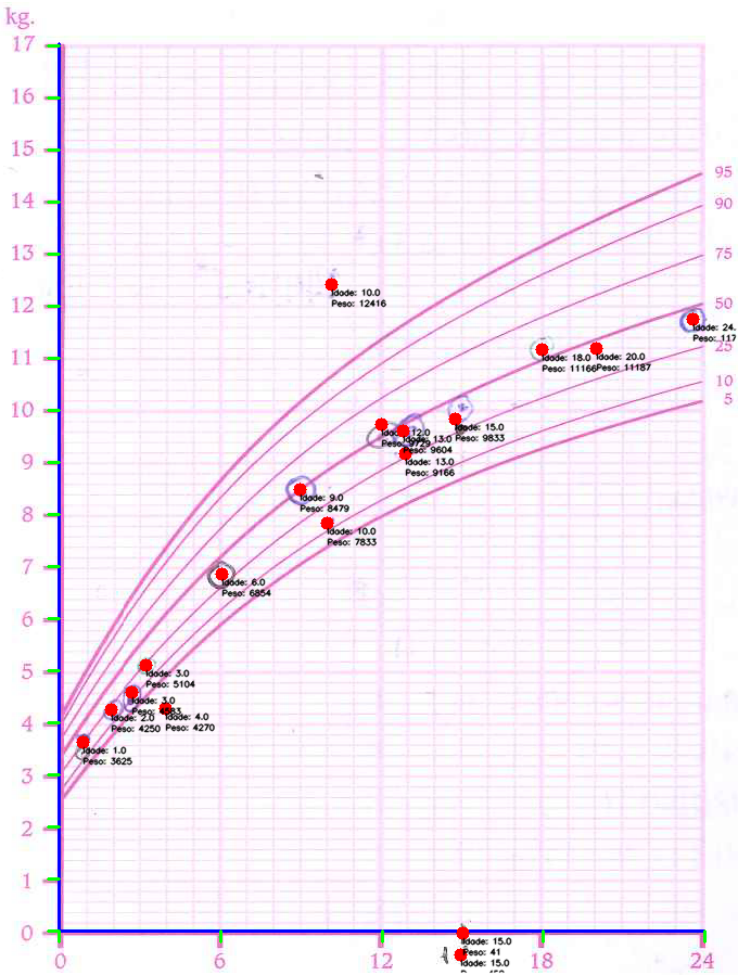



Figure 4.3: Points identified on the chart image

In the chart in question, we can see the points identified in red, as well as the values relative to these points.

In general, practically all points were identified successfully, with a slight deviation from the point marked by the doctor when there are overlapping points.

### 4.2.2 Classifier

To verify the accuracy of each classifier, 5 samples were performed for each dataset. Before each training, a shuffle of the dataset was done to ensure that the sequence of how the data inputted would not affect the output of the classifier. The datasets dimensions are presented in the Table 4.2.

A third dataset was created, in addition to those referenced in 3.2.4, which combines the other two. However, to ensure that there was no predominance for the cases represented by MNIST, a

subset of MNIST that had the same dimensions as the manual dataset was used. This subset was also selected randomly in each of the training iterations performed.

Table 4.2: Datasets dimensions for classifiers

| **Number of** | **MNIST** | **Manual** | **Manual + MNIST** |
|---|---|---|---|
| **Train Images** | 55000 | 450 | 900 |
| **Validation Images** | 5000 | 50 | 100 |
| **Test Images** | 10000 | 100 | 200 |

For each of the classifiers different values were used relative to the numbers of epochs and batch size used. In the Table 4.3 and 4.4 can be seen the values used for the training of the normal neural network and the convolutional neural network, respectively.

Table 4.3: Training arguments size for NN

| | **MNIST** | **Manual** | **Manual + MNIST** |
|---|---|---|---|
| **Epochs** | 100 | 100 | 100 |
| **Batch size** | 100 | 20 | 20 |

Table 4.4: Training arguments size for CNN

| | **MNIST** | **Manual** | **Manual + MNIST** |
|---|---|---|---|
| **Epochs** | 12 | 100 | 100 |
| **Batch size** | 200 | 20 | 20 |

The outcomes are presented in Tables 4.5 and 4.6, where each row corresponds to the results achieved by each training for each dataset. The values presented in these tables are percentages of the hit accuracy in each dataset for their respective test sets. At the bottom of each table are presented, for each dataset, the average values obtained as well as the standard deviation.

The best overall accuracy was reached by the convolutional neural network which obtained **98.96%** for the MNIST dataset, and also obtained higher values than the other model in the other datasets. These results are already expected because it is a more complex model than the neural network with 2 hidden layers

However, it should be noted that for two of the three datasets, the conventional neural network obtained a lower standard deviation, indicating that the variability of hit accuracy is lower.

Table 4.5: Results for NN

| **Train #** | **MNIST** | **Manual** | **Manual + MNIST** |
|---|---|---|---|
| **1** | 98,03% | 84,00% | 83,50% |
| **2** | 98,12% | 81,00% | 86,00% |
| **3** | 98,02% | 85,00% | 85,00% |
| **4** | 97,89% | 84,00% | 86,50% |
| **5** | 98,08% | 82,00% | 87,50% |
| **Average** | 98,03% | 83,20% | 85,70% |
| **Standard deviation** | ± 0,09 | ± 1,64 | ± 1,52 |

Table 4.6: Results for CNN

| **Train #** | **MNIST** | **Manual** | **Manual + MNIST** |
|---|---|---|---|
| **1** | 98,94% | 89,00% | 93,00% |
| **2** | 98,71% | 83,99% | 91,00% |
| **3** | 98,79% | 87,00% | 92,50% |
| **4** | 98,99% | 88,00% | 92,00% |
| **5** | 98,89% | 91,00% | 93,00% |
| **Average** | 98,86% | 87,80% | 92,30% |
| **Standard deviation** | ± 0,11 | ± 2,59 | ± 0,84 |

# Chapter 5

# Conclusions and Future Work

The importance of extracting information from personal health child records has advantages in the fields of medicine, specifically pediatrics. The analysis of this information becomes much faster when it is in digital format, rather than the classic paper format.

In this dissertation, we addressed the problem of extracting the information presented in paper format and converting it to digital format.

In the following sections we compare the objectives achieved with what was eventually planned, as well as the future work needed to improve the solution.

## 5.1 Objectives Fulfillment

During the work made for this thesis, a pipeline was idealized that would perform all necessary procedures to extract the relevant information in the personal child health records and convert it to a digital format, so that it could be stored on a platform where the user could better analyze the medical records of each child.

Several challenges were encountered in isolating and extracting relevant information, such as correctly isolating the digits presented in the table and correctly identifying the points in the chart. These challenges led to the code being implemented for female PCHRs because they were easier to distinguish between pen colors and grid colors.

After extracting the information, an interpretation of this information was necessary. In the case of charts it was a straightforward approach, making it relatively easy to relate the points identified with the respective values to each axis. In the case of the tables, it was necessary to use machine learning to train classifiers that received the images of the isolated characters and to return a value for each one.

Unfortunately, due to lack of time, it was not possible to test the consistency of the extracted information between the tables and the graphs. Also, the platform initially designed to contain all extracted information, ready to be analyzed, was not created. However all the information coming

from the prediction of the characters of the table, as well as the points identified in the graph, are stored in files for future use and creation of the platform.

In short, the pipeline initially planned was not fully completed but the more complex problems were presented for female cases.

## 5.2 Future Work

The ultimate goal was to be able to provide a tool that could extract automatically as much information from as many cases as possible and ask for help from the user when it encountered difficulties. To achieve this goal, some work needs to be done, as well as improvement in some areas.

To complete the pipeline, it is necessary to perform the information consistency between the information extracted from the tables and the charts. After this information has been validated, it is placed on a platform for better interpretation by the user. This platform should allow the user to see the original image and make corrections to the information that is being presented.

As for what has been accomplished of the pipeline, there are stages that can be improved.

Image preprocessing and processing was targeted at female books, requiring a review of the code implemented or rethinking a new way for male cases. For these cases perhaps the approach to a color scheme such as HSV may return better results.

One of the biggest challenges and problems encountered was the segmentation of the characters in each cell, where it was not possible to correctly segment those that overlapped. Here it would be necessary to implement other segmentation methods, as some described in [GCH17].

It would also be interesting to increase the size of the dataset extracted from the digits present in the PCHR provided to see to what extent it would be possible to obtain better results in the training of the classifiers.

With these improvements implemented, it would be possible to achieve a tool that would be able to automate the process of extracting and converting the information present in the personal health child records, which would allow this process that is done manually to be faster and with less costs.

# References


[AG14] Syed Salman Ali and Muhammad Usman Ghani. Handwritten Digit Recognition Using DCT and HMMs. *2014 12th International Conference on Frontiers of Information Technology*, pages 303–306, 2014.

[BA10] Nitin Bhatia and Corres Author. Survey of Nearest Neighbor Techniques. *IJCSIS) International Journal of Computer Science and Information Security*, 8(2):302–305, 2010.

[Bha14] Muzamil Bhat. Digital Image Processing. *International Journal of Scientific & Technology Research*, 3(1):272–276, 2014.

[BL14] Concha Bielza and Pedro Larrañaga. Discrete Bayesian Network Classifiers: A Survey. *ACM Comput. Surv.*, 47(1):5:1—-5:43, jul 2014.

[BMD13] Darío Baptista and Fernando Morgado-Dias. A survey of artificial neural network training tools. *Neural Computing and Applications*, 23(3-4):609–615, 2013.

[CV95] Corinna Cortes and Vladimir Vapnik. Support-Vector Networks. *Machine Learning*, 20(3):273–297, 1995.

[DLS+13] Shifei Ding, Hui Li, Chunyang Su, Junzhao Yu, and Fengxiang Jin. Evolutionary artificial neural networks: A review, 2013.

[FWRZ06] Weiguo Fan, Linda Wallace, Stephanie Rich, and Zhongju Zhang. Tapping the power of text mining. *Communications of the ACM*, 49(9):76–82, sep 2006.

[GCH17] Abdeljalil Gattal, Youcef Chibani, and Bilal Hadjadji. Segmentation and recognition system for unknown-length handwritten digit strings. *Pattern Analysis and Applications*, 20(2):307–323, 2017.

[GL09] Vishal Gupta and Gurpreet S. Lehal. A survey of text mining techniques and applications. *Journal of Emerging Technologies in Web Intelligence*, 1(1):60–76, 2009.

[Gon13] Célia Talma Martins de Pinho Valente Oliveira Gonçalves. *A Tool for Text Mining in Molecular Biology Domains*. PhD thesis, Faculdade de Engenharia da Universidade do Porto, apr 2013.

[GSGN17] Angelos P Giotis, Giorgos Sfikas, Basilis Gatos, and Christophoros Nikou. A survey of document image word spotting techniques. *Pattern Recognition*, 68:310–332, 2017.

[GW02] R. Gonzalez and R Woods. *Digital image processing*. 2002.

[HKP12] Jiawei Han, Micheline Kamber, and Jian Pei. *Data Mining: Concepts and Techniques*. 2012.

[JDM00] Anil K Jain, Robert P W Duin, and Jiangchang Mao. Statistical Pattern Recognition: A Review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:4–37, 2000.

[JJKS16] A. Jailin Reshma, J. Jenushma James, M. Kavya, and M. Saravanan. An overview of character recognition focused on offline handwriting. *ARPN Journal of Engineering and Applied Sciences*, 11(15):9372–9378, may 2016.

[KC16] A. Kaur and D. Chopra. Comparison of text mining tools. *2016 5th International Conference on Reliability, Infocom Technologies and Optimization, ICRITO 2016: Trends and Future Directions*, pages 186–192, 2016.

[LBH+15] Yann LeCun, Yoshua Bengio, Geoffrey Hinton, Lecun Y., Bengio Y., and Hinton G. Deep learning. *Nature*, 521(7553):436–444, 2015.

[LWJ16] Ranyang Li, Hang Wang, and Kaifan Ji. Feature extraction and identification of handwritten characters. *Proceedings - 8th International Conference on Intelligent Networks and Intelligent Systems, ICINIS 2015*, pages 193–196, 2016.

[MAN+14] Gupta Mehul, Patel Ankita, Dave Namrata, Goradia Rahul, and Saurin Sheth. Text-based Image Segmentation Methodology. *Procedia Technology*, 14:465–472, 2014.

[Pat14] Ishani Patel. A Survey on Feature Extraction Methods for Handwritten Digits Recognition. *International Journal of Computer Applications*, 107(12):975–8887, 2014.

[PMR+16] Anisha Priya, Surbhi Mishra, Saloni Raj, Sudarshan Mandal, and Sujoy Datta. Online and Offline Character Recognition : A Survey. pages 967–970, 2016.

[PSH11] J. Pradeep, E. Srinivasan, and S. Himavathi. Diagonal based feature extraction for handwritten character recognition system using neural network. *2011 3rd International Conference on Electronics Computer Technology*, pages 364–368, apr 2011.

[RBD+17] Partha Pratim Roy, Ayan Kumar Bhunia, Ayan Das, Prithviraj Dhar, and Umapada Pal. Keyword spotting in doctor's handwriting on medical prescriptions. *Expert Systems with Applications*, 76:113–128, 2017.

[RLSG16] George Retsinas, Georgios Louloudis, Nikolaos Stamatopoulos, and Basilis Gatos. Keyword Spotting in Handwritten Documents Using Projections of Oriented Gradients. *Proceedings - 12th IAPR International Workshop on Document Analysis Systems, DAS 2016*, pages 411–416, 2016.

[Rod16] Hugo José Freixo Rodrigues. *Ferramenta para Text Mining em Textos completos*. PhD thesis, Faculdade de Engenharia da Universidade do Porto, sep 2016.

[RSCP16] N. Venkata Rao, A.S.C.S Sastry, A.S.N Chakravarthy, and Kalyanchakravarthi P. Optical Character Recognition Technique Algorithms. *Journal of Theoretical and Applied Information Technology*, 2083(2), 2016.

[SF16] Sebastian Sudholt and Gernot A Fink. PHOCNet : A Deep Convolutional Neural Network for Word Spotting in Handwritten Documents. *Proceedings of International Conference on Frontiers in Handwriting Recognition, ICFHR*, pages 1–6, 2016.

REFERENCES

[She00] C Shearer. The crisp-dm model: the new blueprint for data mining. 5:13–22, 01 2000.

[SMS12] Nikita Sharma, Mahendra Mishra, and Manish Shrivastava. Colour Image Segmentation Techniques and Issues : an Approach. *International Journal of Scientific & Technology Research*, 1(4):9–12, 2012.

[TK12] M Thungamani and P Ramakhanth Kumar. A Survey of Methods and Strategies in Handwritten Kannada Character Segmentation. 01(01):18–23, 2012.

[ZA15] Nida M Zaitoun and Musbah J Aqel. Survey on Image Segmentation Techniques. *Procedia - Procedia Computer Science*, 65:797–806, 2015.